

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE

Aula 15

Modelo de caixa (Box model)

Professor: Thiago Theiry de Oliveira

O que já vimos nas primeiras etapas:

- HTML
- CSS
 - Visão geral do CSS.
 - Seletores.
 - Regras visuais.

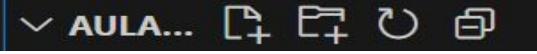


Objetivos da aula.. (Spoilers)

- Continuidade do CSS
 - Modelo de caixa.
 - Box-sizing e border-box
 - Espaçamentos
 - Overflow.
 - Altura e larguras máximas e mínimas.



Configurando o ambiente

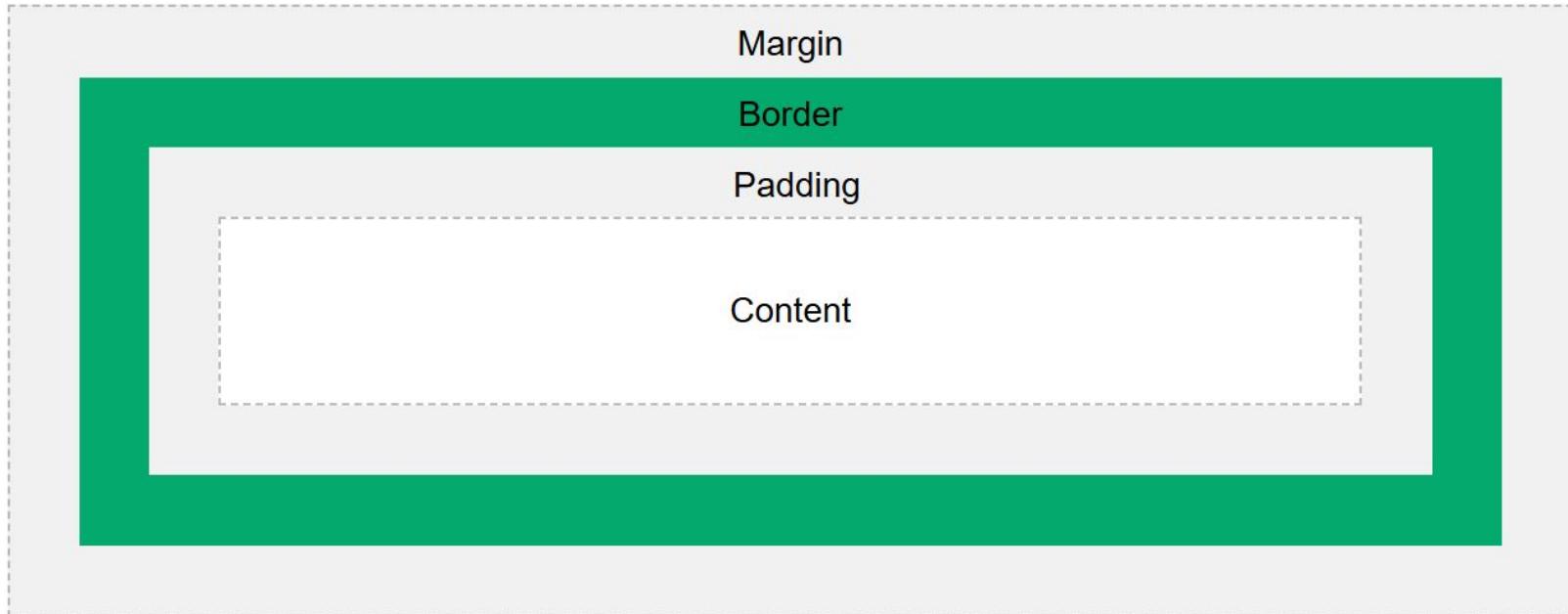
- Iniciando o segundo bimestre, crie uma nova pasta “bimestre3”
- Dentro da pasta aulaPratica01, vamos construir um arquivo de nome css1.html;
 - Selecione o primeiro ícone ao lado 
 - Desde já, TODOS os nossos arquivos DEVEM ter a extensão *.html
 - Use o atalho do vs code para abrir a estrutura padrão do html (pode ser o html simples ou html5)

O modelo de caixa CSS

- Em CSS, o termo "modelo de caixa" é usado quando se fala em design e layout da web.
- O modelo de caixa CSS é essencialmente uma caixa que envolve cada elemento HTML.
- Cada caixa é composta por quatro partes: conteúdo, preenchimento, bordas e margens.
- A imagem a do slide a seguir ilustra o modelo de caixa CSS:

O modelo de caixa CSS

- A imagem a seguir ilustra o modelo de caixa CSS:



O modelo de caixa CSS

- Conteúdo - O conteúdo da caixa, onde aparecem o texto e as imagens
- Padding - Limpa uma área ao redor do conteúdo. O preenchimento é transparente.
- Borda - Uma borda que circunda o preenchimento e o conteúdo
- Margem - Limpa uma área fora da borda. A margem é transparente.

O modelo de caixa CSS

- Demonstração do modelo de caixa:

```
div {  
    width: 300px;  
    border: 15px solid green;  
    padding: 50px;  
    margin: 20px;  
}
```

Largura e altura de um elemento

- Para definir a largura e a altura de um elemento corretamente em todos os navegadores, você precisa saber como o modelo de caixa funciona.

Importante: Ao definir as propriedades de largura e altura de um elemento com CSS, você define apenas a largura e a altura da área de conteúdo . Para calcular a largura e a altura totais de um elemento, você também deve incluir o preenchimento e as bordas.

Largura e altura de um elemento

- Exemplo:

```
div {  
    width: 320px;  
    height: 50px;  
    padding: 10px;  
    border: 5px solid gray;  
    margin: 0;  
}
```

Largura e altura de um elemento

- Aqui está o cálculo:
 - 320px (largura da área de conteúdo)
 - + 20px (preenchimento esquerdo + preenchimento direito)
 - + 10px (borda esquerda + borda direita)
 - = 350px (largura total)
 - 50px (altura da área de conteúdo)
 - + 20px (preenchimento superior + preenchimento inferior)
 - + 10px (borda superior + borda inferior)
 - = 80px (altura total)

Largura e altura de um elemento

- Aqui está o cálculo:
 - A largura total de um elemento deve ser calculada assim:
 - Largura total do elemento = largura + preenchimento esquerdo + preenchimento direito + borda esquerda + borda direita
 - A altura total de um elemento deve ser calculada assim:
 - Altura total do elemento = altura + preenchimento superior + preenchimento inferior + borda superior + borda inferior

Largura e altura de um elemento

- Utilidade:
 - Ele é útil para prever o espaço real que o elemento ocupa.
 - Por exemplo: se você planejou que o card caiba numa coluna de 350px, mas esqueceu de somar padding e border, pode acabar com barra de rolagem horizontal indesejada.

Box-sizing e border-box

- A propriedade CSS box-sizing nos permite incluir o preenchimento e a borda na largura e altura totais de um elemento.
- Por padrão, a largura e a altura de um elemento são calculadas assim:
 - largura + preenchimento + borda = largura real de um elemento
 - altura + preenchimento + borda = altura real de um elemento
- Isso significa que: quando você define a largura/altura de um elemento, o elemento geralmente parece maior do que você definiu (porque a borda e o preenchimento do elemento são adicionados à largura/altura especificada do elemento).

Box-sizing e border-box

- A ilustração a seguir mostra dois elementos <div> com a mesma largura e altura especificadas:

```
div1 {  
    width: 300px;  
    height: 100px;  
    border: 1px solid blue;  
}
```

```
div2 {  
    width: 300px;  
    height: 100px;  
    padding: 50px;  
    border: 1px solid red;  
}
```

Box-sizing e border-box

- A propriedade box-sizing nos permite incluir o preenchimento e a borda na largura e altura totais de um elemento.
- Se você definir box-sizing: border-box; um elemento, o preenchimento e a borda serão incluídos na largura e na altura
- Altere as divs do slide anterior com box-sizing: border-box;

Box-sizing e border-box

- Em resumo o border-box altera a forma como as dimensões de um elemento são calculadas, fazendo com que a largura e a altura definidas incluem o padding (espaçamento interno) e a border (borda), e não apenas o conteúdo

Espaçamento

- Quando falamos de espaçamento, geralmente estamos nos referindo a margem (margin) e preenchimento (padding):
 - Margin: espaço fora do elemento, separando-o de outros elementos.
 - Padding: espaço dentro do elemento, entre o conteúdo (como texto) e a borda do elemento.
- Porém vale ressaltar como é decidido a unidade de espaçamento

Espaçamento - Unidade de medida

- No CSS, o espaçamento pode ser definido usando diferentes unidades. As mais comuns são:
 - Pixels (px)
 - Unidade fixa.
 - $1\text{px} = 1$ ponto na tela (aproximadamente).
 - Simples de entender, mas não escala com o tamanho da fonte do usuário.
 - Exemplo no vs code.

Espaçamento - Unidade de medida

- No CSS, o espaçamento pode ser definido usando diferentes unidades. As mais comuns são:
 - Pixels (px)
 - Unidade fixa.
 - $1\text{px} = 1$ ponto na tela (aproximadamente).
 - Simples de entender, mas não escala com o tamanho da fonte do usuário.

Espaçamento - Unidade de medida

- em
 - Relativo ao tamanho da fonte do elemento pai.
 - $1\text{em} = \text{tamanho da fonte do elemento atual (ou do pai, dependendo do contexto)}$.
 - Útil quando você quer que o espaçamento escale junto com a fonte.
 - Exemplo:

```
p {  
    font-size: 16px;  
    margin-bottom: 2em; /* 2 * 16px = 32px */  
}
```

Espaçamento - Unidade de medida

- rem
 - Relativo ao tamanho da fonte raiz (<html>).
 - $1\text{rem} = \text{tamanho da fonte definida no } <\text{html}>$ (geralmente 16px por padrão).
 - Diferente do em, não depende do pai do elemento, apenas da raiz.
 - Exemplo:

```
html {  
    font-size: 16px;  
}  
  
p {  
    margin-bottom: 1.5rem; /* 1.5 * 16px  
                           = 24px */  
}
```

Espaçamento - Unidade de medida

- %
 - Percentual em relação a algum valor do elemento pai:
 - width ou height → relativo ao tamanho do pai.
 - padding ou margin → geralmente relativo à largura do pai.
 - Útil para layouts responsivos.
 - Exemplo:

```
div {  
    width: 50% /* metade da largura do pai */  
    padding: 5%; /* 5% da largura do pai */  
}
```

Espaçamento - Unidade de medida

- vh e vw
 - vh = 1% da altura da viewport (janela do navegador)
 - vw = 1% da largura da viewport
 - Útil para espaçamentos e tamanhos que dependem da tela.
 - Exemplo:

```
section {  
    padding: 5vh 10vw;  
    /* vertical 5%, horizontal 10% da tela */  
}
```

Overflow

- A propriedade CSS overflow controla o que acontece com o conteúdo que é muito grande para caber em uma área.
- Especifica se o conteúdo deve ser recortado ou se devem ser adicionadas barras de rolagem quando o conteúdo de um elemento for muito grande.

Overflow

- O overflow possui os seguintes valores:
 - visible - Padrão. O estouro não é recortado. O conteúdo é renderizado fora da caixa do elemento.
 - hidden- O estouro é recortado e o restante do conteúdo fica oculto
 - scroll- Barras de rolagem adicionadas. O usuário deve rolar para ver todo o conteúdo.
 - auto- Semelhante a scroll, mas adiciona barras de rolagem somente quando necessário

Altura e larguras máximas e mínimas.

- Em CSS, você pode definir altura (height) e largura (width) de elementos. Porém, às vezes você quer limitar esses valores para que não fiquem muito pequenos ou grandes. Para isso usamos:
 - min-width → largura mínima que o elemento pode ter.
 - max-width → largura máxima que o elemento pode ter.
 - min-height → altura mínima do elemento.
 - max-height → altura máxima do elemento.
- Esses valores ajudam a criar layouts flexíveis, responsivos e evitam quebrar o design.

Recapitulando...

- CSS
 - Modelo de caixa.
 - Box-sizing e border-box
 - Espaçamentos
 - Overflow.
 - Altura e larguras máximas e mínimas.



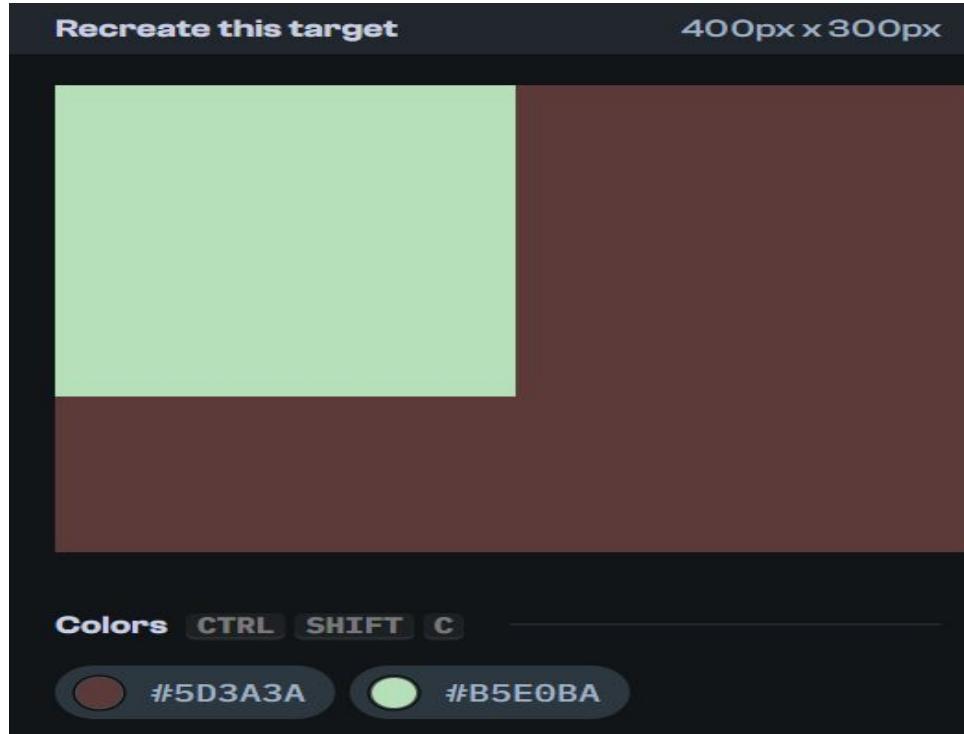
Dúvidas?

Entre em contato: thiago.thierry.17@gmail.com ou sala dos professores



Exercício 1

- Faça o css para gerar essa imagem:



Exercício 2

- Faça o css para gerar essa imagem:



Exercício 3

- Faça o css para gerar essa imagem:

