

# DevWeb

## Capítulo 19

# Imagens de Fundo

Nos capítulos 6 e 11 do nosso material, aprendemos técnicas para usar imagens como parte do nosso conteúdo. Agora, vamos aprender como usar algumas imagens para complementar visualmente um site, aplicando-as aos fundos dos nossos elementos HTML utilizando estilos. Também vamos ver como manter essas imagens adaptáveis ao tamanho do navegador dos nossos visitantes. Vamos nessa?



Você tem todo o direito de usar esse material para seu próprio aprendizado. Professores também podem ter acesso a todo o conteúdo e usá-los com seus alunos. Porém todos o que usarem esse material - seja para qual for a finalidade - deverão manter a referência ao material original, criado pelo **Prof. Gustavo Guanabara** e disponível no endereço do seu repositório público <https://github.com/gustavoguanabara/>. Este material não poderá ser utilizado em nenhuma hipótese para ser replicada - integral ou parcialmente - por autores/editoras para criar livros ou apostilas, com finalidade de obter ganho financeiro com ele.



# O fundo não precisa ter só cor

Em capítulos anteriores, aprendemos a aplicar cores sólidas ou degradê em qualquer elemento de caixa. Mas você não precisa se limitar a essa possibilidade e pode aplicar imagens ao fundo de qualquer elemento exibido visualmente em HTML. Vamos a um exemplo simples com três `<div>` no corpo de um documento:

```
<body>
  <div class="quadrado" id="q1"></div>
  <div class="quadrado" id="q2"></div>
  <div class="quadrado" id="q3"></div>
</body>
```

Agora vamos criar a configuração de estilo base para toda `<div>` que possui a classe `quadrado`:

```
<head>
  <style>
    div.quadrado {
      display: inline-block;
      margin: 5px;
      width: 300px;
      height: 300px;
      background-color: lightgray;
      border: 2px solid black;
      border-radius: 20px;
    }
  </style>
</head>
```

Com esse código aplicado, temos o seguinte resultado na tela:



Viu? Resultado simples, três quadrados exatamente iguais na tela.



**NÃO ENTENDEU?** Se você criou o código acima, mas não obteve o resultado apresentado, é sinal de que talvez você esteja tentando correr demais com seu aprendizado. Se eu puder te dar um conselho, volte ao capítulo 16 e faça todos os exercícios.

Agora vamos adicionar algumas configurações individualmente a cada quadrado, usando os identificadores diferentes entre eles:

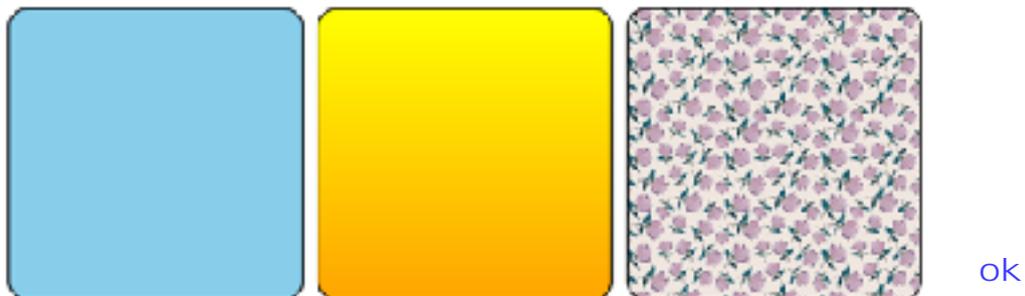
```
div#q1 {  
  background-color: skyblue;  
}  
  
div#q2 {  
  background-image: linear-gradient(to bottom, yellow, orange);  
}  
  
div#q3 {  
  background-image: url('imagens/pattern003.png');  
}
```

Note que na definição da <div> que tem o id com o valor q1, usamos uma cor sólida, já para q2, usamos um preenchimento linear. Já na caixa q3, vamos aplicar uma imagem de fundo através de um endereço informado pela função url().



**CADÊ AS IMAGENS?** Se você quer as imagens que usaremos nesse capítulo, acesse nosso repositório em <https://github.com/gustavoguanabara/html-css/tree/master/exercicios/ex022>. Lá você vai encontrar a pasta com várias imagens que usaremos.

Ao adicionar as linhas acima ao código, o resultado já muda consideravelmente:



De forma geral, essas são as três maneiras mais simples de preencher uma caixa em HTML: cor sólida, degradê ou imagem de fundo.

## E não se esqueça: o body é uma grande caixa

Esse recurso também pode ser usado ao <body> do seu site e o papel de parede será aplicado. Vamos considerar um exemplo bem simples que usa uma imagem disponível no nosso repositório.

```

<style>
  body {
    background-image: url('https://gustavoquanabara.github.io/html-css/imagens/mascote.png');
  }
</style>

```

Analisando o código acima, estamos usando uma `url()` externa, já que a imagem estará em outro servidor (no caso, no servidor do **GitHub**). A imagem em questão é razoavelmente pequena, então não vai ser suficiente para cobrir toda a tela. Sendo assim, o resultado será o seguinte:



## Personalizando a aplicação do background

Quando o tamanho da caixa é maior que o tamanho da imagem, por padrão, a imagem será **repetida nos dois eixos (eixo x e eixo y)** quantas vezes for necessário para cobrir a extensão da caixa contêiner.

É possível alterar esse comportamento usando a propriedade `background-repeat`, que aceita os seguintes valores:

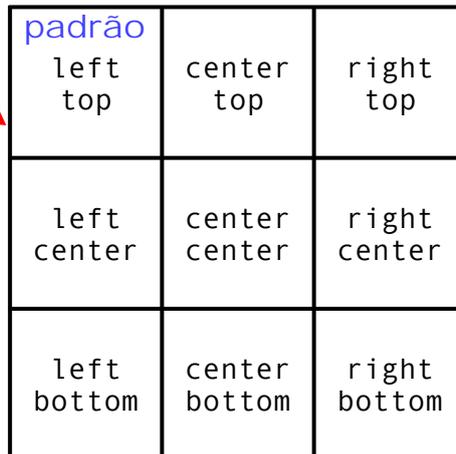
<code>background-repeat: repeat;</code>	
<code>background-repeat: no-repeat;</code>	
<code>background-repeat: repeat-x;</code>	
<code>background-repeat: repeat-y;</code>	

ok

[aula praticada no fundo002.html do exercício 22 - módulo 3](#)

Note que ao usar o `no-repeat`, isso **não obriga** o navegador a aumentar ou diminuir o tamanho da imagem para caber no tamanho da caixa. Para realizar essa adaptação, devemos usar outra propriedade, que veremos mais adiante.

Além de escolher o nível de repetição do `background`, também podemos mudar a **posição de referência** de início das repetições. Por padrão, é considerado o canto esquerdo superior (`left top`), mas podemos ter várias opções. Use a imagem abaixo como referência sempre que precisar definir a posição do fundo com a propriedade `background-position` no seu código.



padrão left top	center top	right top
left center	center center	right center
left bottom	center bottom	right bottom

Outra coisa que podemos fazer é **redimensionar** a imagem para forçá-la a caber na caixa. Por padrão, nenhum redimensionamento será aplicado, e a imagem será exibida do seu tamanho natural. Porém, podemos usar a propriedade `background-size` para alterar esse comportamento.

Os valores aceitos por essa propriedade são:

<code>auto</code>	(padrão) a imagem de fundo será aplicada em seu tamanho original.
<code>[length]px</code> <code>[length]%</code>	Redimensiona a largura da imagem e faz a altura se adaptar automaticamente. Podemos também informar as duas dimensões na sequência ou também usar valores percentuais.
<code>cover</code>	Muda o tamanho da imagem para que ela seja sempre totalmente exibida na tela, <b>sem nenhum corte</b> .
<code>contain</code>	Redimensiona a imagem para que ela cubra o contêiner, mesmo que para isso <b>ocorram alguns eventuais cortes</b> .



**SÓ CONSIGO DEMONSTRAR NA PRÁTICA.** Essas propriedades de personalização de imagens de fundo precisam de demonstração prática. Nesse exato momento, já gravei vários vídeos ensinando o uso de cada uma delas. Em breve você verá, basta acompanhar o canal **Curso em Vídeo no YouTube** 🙌🧐

A última propriedade que podemos configurar é o **vínculo** (`attachment`) da imagem de fundo com o resto do documento, principalmente se o conteúdo for maior do que a altura da página e seja necessário vazar uma rolagem vertical.

A propriedade `background-attachment` aceita os valores:

<code>scroll</code>	(padrão) a imagem de fundo vai rolar junto com o conteúdo.
<code>fixed</code>	A imagem de fundo vai ficar fixada enquanto o conteúdo vai sendo rolado.

## Simplificando as coisas

Assim como já vimos várias vezes em nosso material, existe também a possibilidade de usar uma *shorthand* para simplificar o uso de propriedades que se apliquem ao fundo de uma caixa. A propriedade abreviada `background` pode ser declarada agrupando as seguintes configurações:

- `background-color`
- `background-image`
- `background-position`
- `background-repeat`
- `background-attachment`

Sendo assim, no lugar de usar:

```
background-color: ■black;
background-image: url('imagens/wallpaper002.jpg');
background-position: center center;
background-repeat: no-repeat;
background-attachment: fixed;
```

Podemos reunir tudo em uma única declaração:

```
background: ■black url('imagens/wallpaper002.jpg') center center no-repeat fixed;
background: black url(/imagens/wallpaper002.jpg) center center no-repeat fixed;
background-size: cover; --- o size não funciona no shorthand !!!
```

## Centralização vertical em contêineres

Antes de começar a explicar o assunto sobre o qual vamos falar, preciso desabafar: plural de *contêiner* é muito esquisito, não acha? A propósito, o significado de *contêiner* (versão do Inglês *container*) é simples e direto: "aquele que contém coisas".



Aprendemos no **capítulo 16** que existem elementos que podem conter outros elementos. As `<div>` são um exemplo de elemento *contêiner*. Quando queremos centralizar blocos horizontalmente, aprendemos a usar o `margin:auto`; nas folhas de estilo. Mas como fazer a centralização vertical?

No **capítulo 17**, onde criamos o nosso primeiro mini-projeto, tivemos que arrumar uma maneira de centralizar e redimensionar um vídeo dentro de um *contêiner*. Agora vou te mostrar uma outra técnica.

Vamos começar criando uma hierarquia simples entre dois blocos:

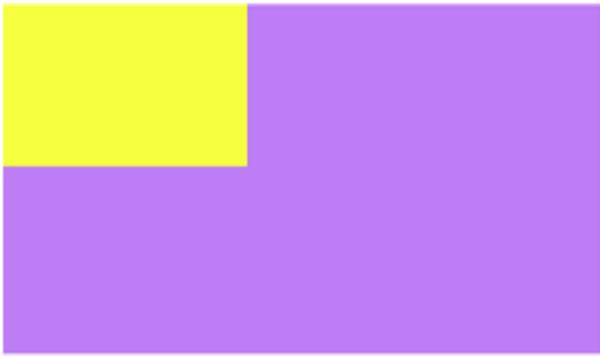
```
<div id="fora">
  <div id="dentro">
  </div>
</div>
```

Agora vamos criar as configurações personalizadas para cada <div> dentro da área de <style> na área da cabeça <head> do código:

```
<style>
  div#fora {
    height: 96vh;
    background-color: #8D7DF5;
  }

  div#dentro {
    height: 200px;
    width: 300px;
    background-color: #F5FF40;
  }
</style>
```

O resultado disso será algo como:



O nosso objetivo aqui é deixar o retângulo interno exatamente no meio do retângulo externo. Vamos começar configurando o posicionamento de cada uma. O retângulo externo terá posicionamento relativo, enquanto o interno terá posicionamento absoluto.

```
div#fora {
  position: relative;
  ...
}

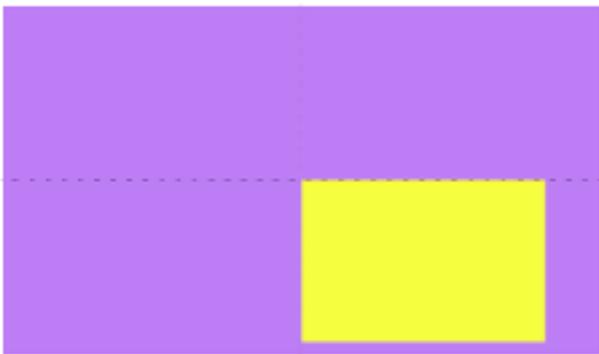
div#dentro {
  position: absolute;
  ...
}
```

Não crie um novo seletor, apenas adicione as declarações que estou indicando. Qualquer dúvida, assista o último vídeo relativo ao **capítulo 19** do curso.

Quando definimos um bloco com posicionamento absoluto, podemos personalizar a sua posição exata através das propriedades `left` e `top`. Como queremos posicionamento centralizado, vamos configurar os dois na metade do contêiner:

```
div#dentro {  
  position: absolute;  
  
  left: 50%;  
  top: 50%;  
  ...  
}
```

Porém, infelizmente, o resultado não será exatamente o que esperamos. Mas não se desespere! Nem tudo está perdido!



Note que, pelas linhas pontilhadas, o retângulo interno foi realmente posicionado a 50% da tela, mas pelo canto superior esquerdo da caixa. Vamos realizar uma transformação e mover o retângulo interno para a esquerda e para cima, para que ele fique efetivamente centralizado.

```
div#dentro {  
  ...  
  transform: translate(-50%, -50%); transform: translate(-50%, -50%)  
  ...  
}
```

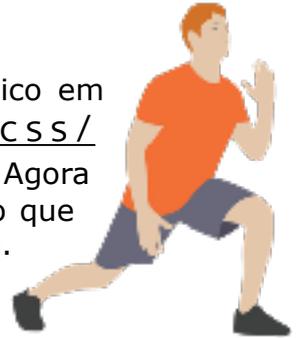
valor negativo



E está feito! Essa é apenas uma das técnicas de centralização de conteúdo, mas as outras requerem aprender outros conceitos mais aprofundados das folhas de estilo, como as caixas flexíveis (*Flexbox*).

# Hora de exercitar

Chegou a hora de acessar o endereço do nosso repositório público em <https://gustavoguanabara.github.io/html-css/desafios/> e executar os **exercícios 022** no seu computador. Agora tente atingir esse mesmo resultado em casa, sem copiar o código que eu criei. Nesse momento, a prática é algo que você mais precisa. Se por acaso ficar difícil, pode acessar o repositório público de HTML e CSS e dar uma olhada nos comandos, mas **EVITE COPIAR**.



## Tenho desafios pra você!

Lá no repositório, além do material em PDF e dos códigos dos exercícios 100% disponíveis, também disponibilizamos alguns **desafios** que devem ser resolvidos. Esses desafios não incluem o código original e você deve tentar chegar à resposta sem copiar nenhum código.

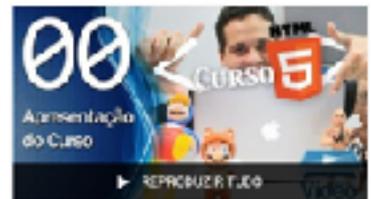


Com todo o conteúdo que vimos até essa aula, você já pode resolver o **desafio d011**. Acesse o repositório público, abra a área do curso de HTML+CSS e clique no link de acesso aos desafios. Manda ver! Só não fica pedindo a resposta! Você consegue resolver isso sozinho(a)!

Repositório em: <https://gustavoguanabara.github.io>

## Eu já falei sobre isso no YouTube?

Eu sei que às vezes as pessoas gostam mais de assistir vídeos do que ler livros, e é por isso que eu lanço há anos materiais no canal Curso em Vídeo no YouTube. O link que vou compartilhar contigo tem o conteúdo explicado como você leu aqui, só que de forma mais ilustrada. Reserve um tempo dos seus estudos para assistir esse vídeo todo.



Curso em Vídeo: [https://www.youtube.com/playlist?list=PLHz\\_AreHm4dlAnJ\\_jJtV29RFxnPHDuk9o](https://www.youtube.com/playlist?list=PLHz_AreHm4dlAnJ_jJtV29RFxnPHDuk9o)

## Teste seus conhecimentos

Terminou de ler esse capítulo e já acompanhou todos os vídeos e referências externas que indicamos? Pois agora, responda a essas 10 perguntas objetivas e marque em cada uma delas a única opção verdadeira. Aí sim, você vai poder comprovar que realmente entendeu o conteúdo.



Questão... sem questões ....

- 
- 
- 
- 

Questão...

- 
- 
- 
-

Questão...

- 
- 
- 
- 

Questão...

- 
- 
- 
-