



**INSTITUTO FEDERAL**  
Sul-rio-grandense

Câmpus  
Pelotas

# LPW - Linguagem de Programação para WEB

## CSS

### Aula 7

Professor Sérgio Rodrigues



# CSS

## Trabalhando com Imagens

# Inserir Imagens

Duas formas para inserirmos imagens:

## HTML

```
<img src="" alt="">
```

## CSS

```
background-image: url(...);
```

# A Tag img

Imagens são o ponto forte na Web. Galerias de fotos, imagens de fundo e decorativas, são inúmeras as aplicações.

Uso do elemento img? Para inserir uma imagem que *tem relevância para o conteúdo da página*. Imagens meramente decorativas devem ser inseridas através de CSS, com a propriedade background, que veremos a seguir:

# A Tag img

Uma Tag img é escrita desta forma:

```
src="smiley.gif" alt="Smiley face" height="42" width="42">
```

A tag img não possui conteúdo, assim como as tags link, meta e br. Portanto apenas trabalhamos com seus atributos e não há tag separada de fechamento.

# A Tag img

## Atributos Tag img:

- src ("source", ou origem);
- width (largura) e height (altura) em pixels (mas é possível utilizar um valor de porcentagem, por exemplo). Obs.: Não são atributos obrigatórios, portanto podemos omiti-los e controlar as dimensões com CSS ou JavaScript;
- alt é de extrema importância por questões de acessibilidade e também motores de busca tipo Google usam essa informações para ranking.

# A Tag img

Dica Tag img:

Se a ideia para a descrição alt de uma imagem for algo como "Imagem", "Enfeite" ou simplesmente "", provavelmente você não deveria estar usando o elemento img;

A tag img adiciona *conteúdo* à página e não tem apenas fins de decoração visual;

Para imagens decorativas, usamos CSS como, por exemplo, com a propriedade background.

# A Propriedade background

Com a propriedade CSS background podemos colocar uma imagem de fundo em qualquer elemento de nossa página; Também podemos controlar a direção de repetição, posição, entre outros aspectos.



# APLICANDO UMA COR DE FUNDO

Para colocar uma cor de fundo em um elemento utilizamos o background-color. Vamos também colocar um pouco de padding (margem interna) e estilos simples de fonte na classe a seguir:

```
.fundo-imagem {  
    background-color: #FFC; /* equivale a #FFFFFFCC */  
    padding: 1em;  
    font-family: sans-serif;  
    font-size: 14px;  
}
```

Para especificar a cor, podemos usar qualquer um dos formatos que já vimos, como hexadecimal, RGB, RGBA, palavras-chave, entre outros.

# APLICANDO UMA IMAGEM DE FUNDO

Para inserir uma imagem de fundo, utilizamos `background-image`:

```
<style>
.fundo-imagem {
    background-color: #FFC;
    background-image: url(imagens/erro.jpg);
    padding: 7em;
    font-family: sans-serif;
    font-size: 15px;
}
</style>
```



exe7\_2

# APLICANDO UMA IMAGEM DE FUNDO

A imagem de fundo foi aplicada, mas ela está se repetindo indefinidamente. Às vezes este é o comportamento desejado, mas não para o que queremos fazer neste exemplo. Para controlar a repetição, usamos background-repeat, com valores que incluem:

- repeat: a imagem repete nos eixos horizontal e vertical, ou x e y;
- repeat-y: a imagem repete no eixo vertical, ou y;
- repeat-x: a imagem repete no eixo horizontal, ou x;
- no-repeat: a imagem não repete.

# APLICANDO UMA IMAGEM DE FUNDO

Posição da imagem usa o background-position. Esta propriedade aceita dois valores: um para x e um para y, coordenadas estas a partir do canto superior esquerdo do container (neste caso, o parágrafo).

Podemos escrever os valores de background-position de várias maneiras (sempre um valor para o eixo x e outro para y, nesta ordem):

# APLICANDO UMA IMAGEM DE FUNDO

- Através de números exatos, em pixels, em, ou outra unidade. Ex: background-position: 10px 15px; posicionará a imagem a 10 pixels da esquerda e a 15 pixels do topo do elemento. É possível também utilizar valores negativos, que movem a imagem de fundo na outra direção.
- Através de porcentagens. Ex: background-position: 50% 50%; irá posicionar a imagem de fundo no centro do elemento.
- Utilizando palavras-chave, como: top, left, right, bottom e center. Ex: background-position: right bottom irá posicionar a imagem no canto inferior direito do elemento.

# APLICANDO UMA IMAGEM DE FUNDO

Podemos juntar todas as declarações em uma só, com a propriedade background. A sintaxe é esta:

```
background: [cor] [imagem] [repetição] [posição]</code>
```

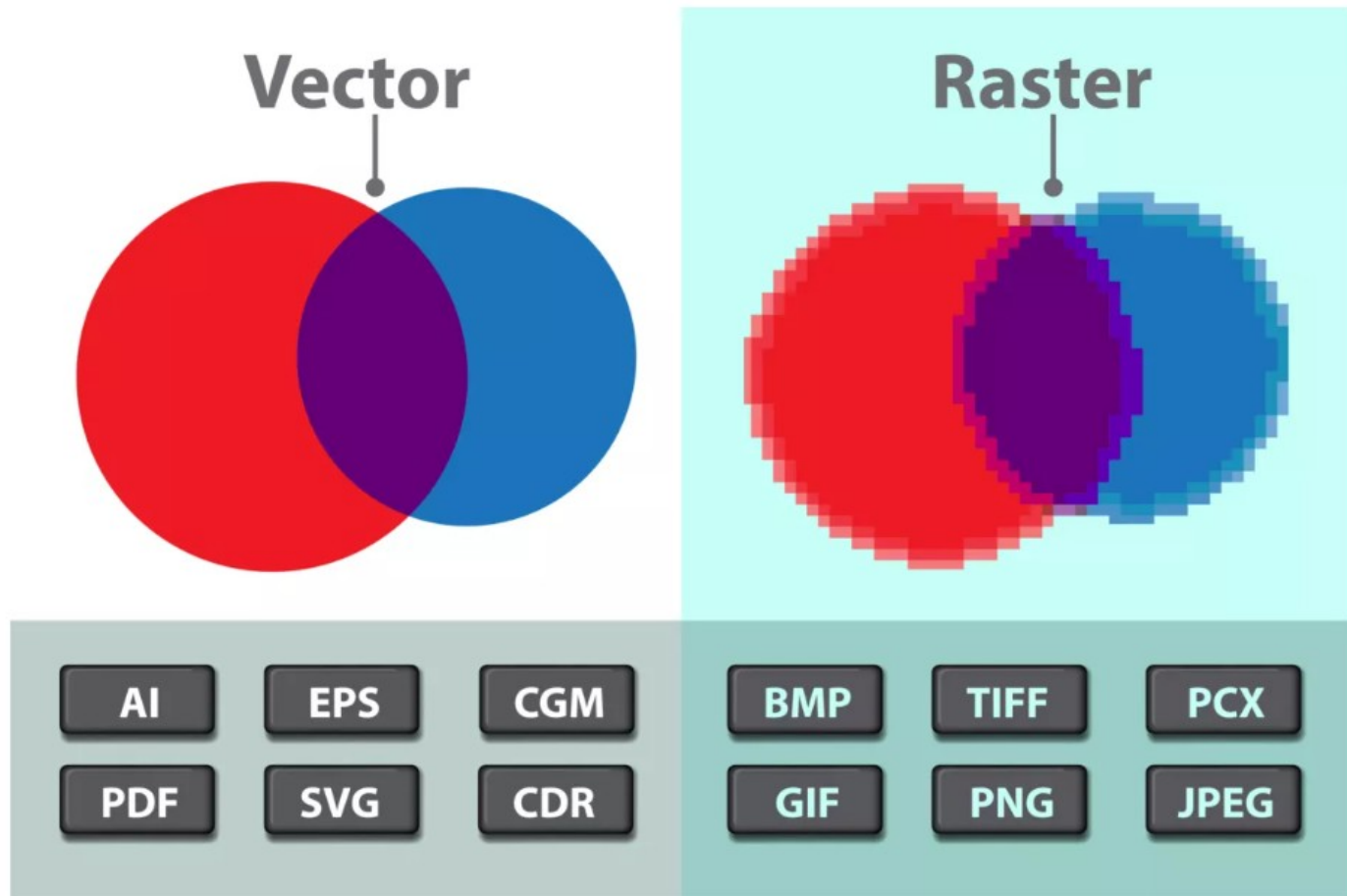
Veja:

```
.fundo-imagem {  
    background: #FFC url(error.png) no-repeat 10px 50%;  
    padding: 1em 1em 1em 2.5em;  
    font-family: sans-serif;  
    font-size: 14px;  
}
```

# APLICANDO UMA IMAGEM DE FUNDO

- A propriedade background-attachment define se uma imagem de plano de fundo rola (scroll) com o restante da página ou é fixa (fixed).
- A propriedade background-size especifica o tamanho das imagens de fundo. Valores (auto, contain, cover, %, px)
- Observe as diferenças [background-size](#)

# Tipos de Imagens



Novos formatos : webp e avif

<https://caniuse.com/>



# Tipos de Imagens

## Raster/Bitmap

- Formada por pixels;
- Cada pixel com uma cor e tonalidade;
- Uso fotografias.

## Vector

- Cálculos matemáticos (coordenadas x e y);
- Mais leve, maior compressão;
- Não perde resolução (zoom).

# A PROPRIEDADE POSITION

As propriedades CSS **position** e **float** , são os meios mais tradicionais de se construir layouts de websites.

Utilizamos position para posicionar elementos utilizando coordenadas específicas, através das propriedades top (topo), left (esquerda), right (direita) e bottom (baixo).

A propriedade float foi trabalhada na aula anterior.

# A PROPRIEDADE POSITION

Os principais valores da propriedade position são:

- **static:** é o valor padrão. O elemento é posicionado de acordo com a sua ordem no código fonte.
- **absolute:** retira o elemento do fluxo do documento (como se ele não existisse na página). O elemento é posicionado segundo as coordenadas indicadas, estas por sua vez relativas ao canto superior esquerdo da página ou do contexto mais próximo.

# A PROPRIEDADE POSITION

Propriedade position (continuação):

- **relative:** o elemento é posicionado relativamente à sua posição atual. Mesmo saindo da posição atual, o espaço que o elemento ocupava no documento continua sendo ocupado e contabilizado no fluxo de elementos.
- **fixed:** o elemento é fixado na página de acordo com as coordenadas especificadas (relativas ao canto superior esquerdo da página). Ele é retirado do fluxo do documento e sempre fica na mesma posição definida, não importando o tamanho da janela ou rolagem da mesma.

exe 7\_5 6,7,8

# CONTEXTO DE POSICIONAMENTO

Definir e utilizar **contextos de posicionamento** é uma das técnicas mais comuns na construção de websites, pois oferece precisão e confiabilidade no posicionamento e ampla compatibilidade entre navegadores.

Como exemplo, iremos construir um cabeçalho bem simples de um site. Para isso, iremos usar duas divs, uma com a classe "cabecalho" e a outra com a classe "logo":

Ver EXEMPLO:

# ATIVIDADE

Pesquise no W3schools as seguintes propriedades:

- 1) position: sticky
- 2) z-index
- 3) overflow

# TABELAS

Tabelas são elementos básicos do HTML. Muitos websites já foram feitos utilizando-se somente tabelas para toda a construção de seus layouts.

Com o avanço do CSS e dos *Web Standards*, dispomos de ferramentas que resolvem a questão de layout de uma forma mais satisfatória.

Em uma tabela HTML, primeiro criamos as linhas, com a tag `tr` (*table row* ou linha de tabela), e depois as colunas internas, com a tag `td` (*table division* ou divisão de tabela).

# TABELAS

```
<table>
  <tr>
    <td>
      Nome
    </td>
    <td>
      Telefone
    </td>
    <td>
      E-mail
    </td>
  </tr>
</table>
```



# TABELAS

Podemos melhorar ainda mais a acessibilidade desta tabela com o atributo scope. Este atributo serve para identificar a relação entre uma th e as colunas ou linhas subsequentes. Se o título descreve linhas abaixo, utilizamos col; se o título descreve as células ao lado, usamos row.

```
<table>
  <caption>Lista de Contatos</caption>
  <tr>
    <th scope="col">
      Nome
    </th>
    <th scope="col">
      Telefone
    </th>
    <th scope="col">
      E-mail
    </th>
  </tr>
```

# TABELAS

Podemos mesclar linhas ou colunas de uma tabela com os atributos rowspan e colspan, respectivamente, nos elementos th e td.

```
<table>
  [...]
  <tr>
    <th scope="row" colspan="4">Total</th>
    <td>R$100,00</td>
  </tr>
</table>
```

# TABELAS

Há alguns anos era muito comum colocar alguns atributos na tag table, como cellpadding="0" cellspacing="0" border="0". Com CSS, algumas simples declarações eliminam esses atributos hoje desnecessários:

```
.compra {  
    border-spacing: 20;  
    border-collapse: separate;  
}  
  
.compra th, .compra td {  
    border: 1px solid #F30;  
    padding: 3px;  
}
```

A vantagem em retirar estes padrões de espaço e bordas é que podemos construir uma apresentação visual mais consistente.

# TABELAS

A propriedade text-align

Propriedade para alinhamento (left, right, center)

```
table, td, th {  
    border: 1px solid black;  
}  
table {  
    border-collapse: collapse;  
    width: 100%;  
}  
td {  
    text-align: center;
```

# TABELAS

## Tabela Flutuante

Usando pseudo-classes (:hover) sobre a tabela

```
table {  
    border-collapse: collapse;  
    width: 100%;  
}  
th, td {  
    padding: 8px;  
    text-align: left;  
    border-bottom: 1px solid #ddd;  
}  
tr:hover {background-color: coral;}
```

# TABELAS

## Tamanho da Tabela

Para definir a largura de uma tabela, adicione o atributo style ao elemento <table>

```
<table style="width:100%">
  <tr>
    <th style="width:70%">First Name</th>
    <th>Last Name</th>
    <th>Age</th>
  </tr>
  <tr>
```