CSS
Trabalhando com Imagens

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:

```
<img 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 rancking.

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 pagina 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 #FFFFCC */
    padding: 1em;
    font-family: sans-serif;
    font-size: 14px;
}
```

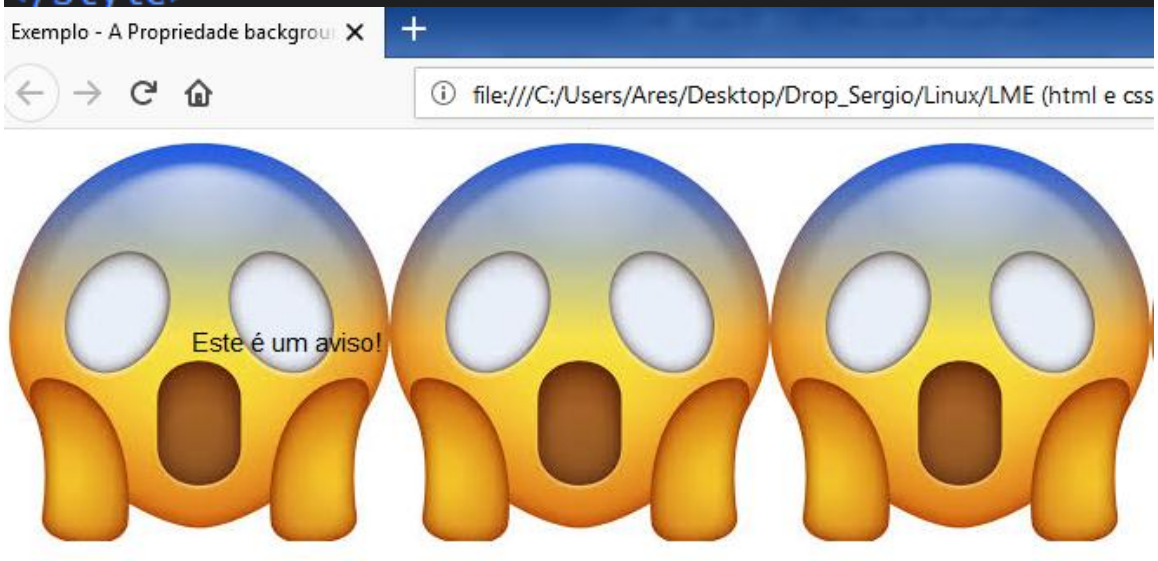
Para especificar a cor, podemos usar qualquer um dos formatos que já vimos, como hexadecimal, RGB, RGBA, palavras-chave, entre outros.

exe7_1

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.

exe7_3

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 o 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, botton e center. Ex: background-position: right bottom irá posicionar a imagem no canto inferior direito do elemento.

exe7_4

APLICANDO UMA IMAGEM DE FUNDO

Por fim, 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;
}
```

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 passada.

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.
- 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:

exe7_9

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>
```

Exe7_10

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</td>
        <td>R$100,00</td>
    </tr>
</table>
```

Exe7_11

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.

exe7_12

ATIVIDADE

1) Crie uma tabela qualquer como 5 colunas e 8 linhas faça estilização CSS.
2) Procure na Internet ao menos 3 (três) exemplos de tabela com estilização CSS olhe o código fonte e faça algumas alterações na tabela criada anteriormente.
3) Anote as propriedades/elementos e valores/atributos usados nas estilizações.

FORMULÁRIOS

Revisando todo formulário começa com uma tag <form>.

```
<form action="#" method="post"></form>
```

Para agrupar todos os campos do formulário, usa-se a tag <fieldset> (grupo de campos).

```
<form action="#" method="post">
    <fieldset>
        <!-- campos -->
    </fieldset>
</form>
```

FORMULÁRIOS

Para mais versatilidade e facilidade na criação das CSS, é recomendável englobar o rótulo do campo (label) e o campo com um outro elemento.

```
<div class="campo">
    <label for="nome">Nome</label>
    <input type="text" name="nome" id="nome">
</div>
```

A tag label é a tag mais apropriada para justamente "rotular" os campos de um formulário. O seu atributo **for** serve para associar o rótulo ao campo desejado. A vantagem de ter as divs com a classe "campo" ficará mais evidente quando fizermos a estilização CSS do formulário.

FORMULÁRIOS

A tag fieldset que vimos antes também será utilizada para agrupar campos lado a lado, através da classe "grupo":

```
<fieldset class="grupo">
    <div class="campo">
        <label for="nome">Nome</label>
        <input type="text" name="nome" id="nome">
    </div>
    <div class="campo">
        <label for="snome">Sobrenome</label>
        <input type="text" name="snome" id="snome">
    </div>
</fieldset>
```

FORMULÁRIOS

A tag input é extremamente versátil. Com o atributo type, conseguimos configurá-la de várias formas diferentes. Se queremos um campo simples de texto, escrevemos:

```
<input type="text" name="nome" size="30" maxlength="30">
```

Outros atributos:

minlength: número mínimo de caracteres deve ser preenchido;
required: indica campo de preenchimento obrigatório;
value: indica o que ficará preenchido no campo por padrão;
disabled: desabilita o campo, evitando o seu preenchimento.

FORMULÁRIOS

Caso queiramos um campo de envio de arquivos, escrevemos:

```
<input type="file" name="arquivo" size="30">
```

Para campos de escolha de alternativas, temos `radio` e `checkbox`:

```
<input type="radio" name="radio"> Escolha 1
<input type="radio" name="radio" checked> Escolha 2
<input type="radio" name="radio"> Escolha 3
<br>
<input type="checkbox" name="checkbox"> Escolha 1
<input type="checkbox" name="checkbox" checked> Escolha 2
<input type="checkbox" name="checkbox"> Escolha 3
```

FORMULÁRIOS

Para botões tipo "radio" e "checkbox", uso um label com a classe "checkbox". Assim, todo o campo, incluindo o rótulo e o botão, fica automaticamente clicável, melhorando a sua usabilidade (neste caso não precisamos do atributo for):

```
<label class="checkbox">
    <input type="radio" name="sexo" value="masculino"> Masculino
</label>
<label class="checkbox">
    <input type="radio" name="sexo" value="feminino"> Feminino
</label>
```

exe7_14

FORMULÁRIOS

A tag <button>, essa tag é mais fácil de trabalhar e de aplicar estilos que, por exemplo, uma tag input tipo "submit". Suas vantagens incluem uma aparência mais consistente entre navegadores e a possibilidade de inserir conteúdo variado entre as tags de abertura e fechamento (ex: texto e imagens).

```
<button type="submit" class="botao submit">Enviar</button>
```

Veja o exemplo completo:

exe7_15

FORMULÁRIOS

Aplicar CSS a formulários não é muito diferente de trabalhar com outros elementos já vistos em aula.

```css
* {
    margin: 0;
    padding: 0;
}

fieldset {
    border: 0;
}

body, input, select, textarea, button {
    font-family: sans-serif;
    font-size: 1em;
}
```

Estilo reset (*) zerar as margens, além de retirar a borda padrão dos fieldsets. Também padronizo a fonte e o tamanho de fonte dos campos.

FORMULÁRIOS

Melhorando o formulário.

```
.campo {
    margin-bottom: 1em;
}

.campo label {
    margin-bottom: 0.2em;
    color: #666;
    display: block;
}                   /*Como todos os campos estão agrupados dentro destes elemen
                        estilos como margens de forma uniforme */

fieldset.grupo .campo {
    float:  left;
    margin-right: 1em;
}                          /* Flutuamos os campos lado a lado Cidade e Estado */
```

Exercícios