

PRÁTICA LABORATORIAL 04

Objetivos:

- Constituição de um ficheiro CSS3
- Interligação do CSS com o HTML
- Diferença entre id e class
- Cores
- Fundos
- Margin e Padding
- Width e Height
- Realçar que a explicação teórica e prática está disponível em vídeo em:
 - Parte 1: https://youtu.be/o0Y_TGmJhW4
 - Parte 2: <https://youtu.be/3lWI5K1PzhA>
 - Parte 3: https://youtu.be/_3e8e-yayE

Cascading Style Sheets ou mais vulgarmente denominado por CSS é usado para estilizar elementos escritos numa linguagem de marcação como o HTML.

PARTE I – INTEGRAÇÃO DO CSS COM O HTML

Há três formas de integrarmos CSS num ficheiro HTML:

- Num ficheiro de CSS associado a um ou mais ficheiros de HTML;
- Dentro do <head> do ficheiro HTML;
- Dentro da tag do elemento HTML a aplicar.

Para exemplo vamos criar um ficheiro HTML com um conteúdo bastante simples:

- Um elemento <h1> com o conteúdo Página de HTML com CSS;
- Um elemento <hr>;
- Dois elementos <p> com parágrafos gerado por Lorem.

Vamos criar um ficheiro com o nome “estilo.css” e referenciar esse mesmo ficheiro no ficheiro HTML. Dentro do <head> do ficheiro HTML, colocar: <link rel="stylesheet" href="estilo.css">

No ficheiro CSS, adicione o seguinte código:

```
body {  
  background-color: lightgrey;  
  padding: 20px;  
  text-align: center;  
}
```

Este código irá aplicar a todo o conteúdo do <body> uma cor de fundo (background-color) cinzenta-clara, uma margem interna (padding) de 20 pixels e alinhar o conteúdo ao centro.

Todas as formatações indicadas no seletor body estão a ser aplicadas a todo o conteúdo presente no body, ainda assim podemos especificar o elemento ou elementos a que queremos aplicar a formatação, por exemplo, retire a propriedade text-align do body e insira um novo seletor p com essa característica:

```
body {  
  background-color: lightgrey;  
  padding: 20px;  
}  
p {  
  text-align: center;  
}
```

PARTE II – ID VS CLASS

Neste caso o alinhamento ao centro foi aplicado apenas aos parágrafos, permitindo assim especificar o estilo a todos os elementos do tipo p.

Vamos considerar que queremos fazer alinhamentos diferente ao longo da página: o <h1> será alinhado ao cento, o primeiro <p> à esquerda e o segundo <p> à direita.

Temos aqui um problema que é termos dois elementos com o mesmo seletor e estilos diferentes, aqui entram os ID.

No código HTML adicione ao primeiro parágrafo a propriedade:

```
id = "alinhaEsquerda"
```

E no segundo:

```
id = "alinhaDireita"
```

Altere o código CSS para a formatação ter efeito, assim:

```
body {  
  background-color: lightgrey;  
  padding: 20px;  
}  
h1{  
  text-align: center;  
}  
#alinhaEsquerda {  
  text-align: left;  
}  
#alinhaDireita {  
  text-align: right;  
}
```

Os seletores de CSS definidos pelo utilizador podem ser agrupados em dois tipos:

- Identificadores: no HTML têm a propriedade id e no CSS são representados com o símbolo # antes do nome atribuído. São usados em apenas um elemento.
- Classes: no HTML têm a propriedade class e no CSS são representados com o símbolo . antes do nome atribuído. São usados em apenas um ou mais elementos.

Acrescente dois elementos <h2> antes de cada parágrafo com o conteúdo “Parágrafo 1” e “Parágrafo 2”. Atribua a ambos a propriedade class=“alinhaCentro”. Crie o CSS de acordo com a propriedade definida.

Podemos também aplicar a vários elementos de uma só vez um determinado estilo, para isso colocamos todos os seletores separados por vírgula antes das propriedades. Exemplo:

```
h1, h2, h3, p{  
  text-align: center;  
}
```

Irá aplicar o alinhamento ao centro para todos os h1, h2, h3 e p.


É também possível especificar que queremos adicionar um determinado estilo a um elemento que se encontre dentro de um bloco específico, por exemplo, teria um <h1> dentro do bloco <header> e um <h1> dentro do bloco <main>, posso definir o seguinte:


index.html

estilo.css

index.html (browser)

```
<header>
  <h1>Texto</h1>
</header>
<main>
  <h1>Texto</h1>
</main>
```

```
header h1{
  color:  red;
}

main h1{
  color:  blue;
}
```



NOTA: Os comentários em CSS são feitos iniciando com / e terminando com */*

PARTE III – CORES

Como já vimos anteriormente uma das propriedades de CSS que podemos aplicar é a alteração de cores. Para escrita das mesmas podem ser utilizadas três notações:

- Nome da cor (lista aqui: https://www.w3schools.com/colors/colors_names.asp);
- Código hexadecimal;
- Código RGB.




Crie as três classes CSS seguintes e aplique-as a três elementos na página à sua escolha:

```
.vermelho{  
    color:  red;  
}  
.azul{  
    color:  #00F;  
}  
.verde{  
    color:  rgb(0, 255, 0);  
}
```

NOTA: no Visual Studio Code, colocando o cursor em cima da cor é possível selecionar a cor na paleta de cores que é apresentada.

Há vários parâmetros que podem ser utilizados com cor, nos elementos acima o alterado será a cor do texto, mas também podemos alterar:

- Cor de fundo;
- Cor do limite do elemento – borda.

```
.vermelho{  
    color:  red;  
    background-color:  lightseagreen;  
    border: 2px solid  black;  
}
```



É ainda possível alterar a transparência da cor acrescentando à cor RGB o parâmetro alpha. De notar que a cor passa a estar definida como rgba em vez de rgb. O valor do alpha varia entre 0.0 (totalmente transparente) e 1.0 (totalmente opaco).

color: rgba(255, 0, 0, 0);



color: rgba(255, 0, 0, 0.5);



color: rgba(255, 0, 0, 1);



PARTE IV – FUNDOS

Relativamente aos fundos já conseguimos ver que aplicando o `background-color` conseguimos alterar a cor de fundo. Associada a esta propriedade podemos também definir a propriedade `opacity` para identificar tal como o alpha no texto, a transparência do fundo.

Podemos ainda aplicar uma imagem como fundo da página.

`background-image: url("imagem.jpg");`


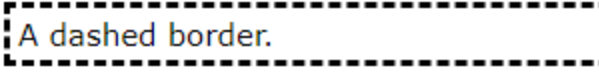


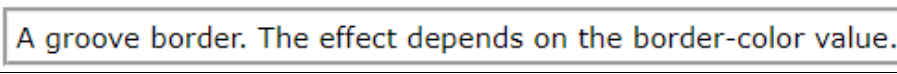
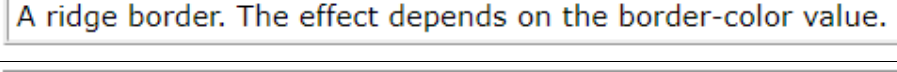
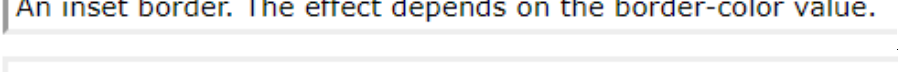
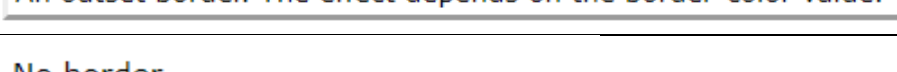
É ainda possível associar as seguintes propriedades:

- `background-repeat: repeat-x;` → repete a imagem na horizontal;
- `background-repeat: repeat-y;` → repete a imagem na vertical;
- `background-repeat: no-repeat;` → não repete a imagem;
- `background-position: right top;` → posiciona a imagem no cimo à direita da página;
- `background-attachment: fixed;` → mantém a imagem fixa quando deslizamos na página;
- `background-attachment: scroll;` → a imagem desliza enquanto navegamos na página.

Existem ainda outras propriedades e uma forma condensada de escrever as propriedades, algo que pode ser analisado em: https://www.w3schools.com/css/css_background_shorthand.asp


PARTE V – LIMITES

Os limites, ou borders, também podem ser definidos utilizando atributos, começando pelo border-style que aceita os seguintes tipos:

dotted	
dashed	
solid	
double	
groove	
ridge	
inset	
outset	
none	No border.
hidden	A hidden border.

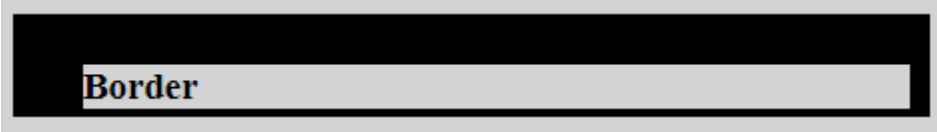
É ainda possível aplicar estilos mistos como no exemplo seguinte:

```
p {  
    border-style: dotted dashed solid double;  
}
```



A espessura do traço é também um parâmetro personalizável utilizando o border-width (normalmente indicado em pixels – px). Tal como no border-style é possível definir espessuras diferentes para os 4 limites do elemento:

border-width: 25px 10px 4px 35px;




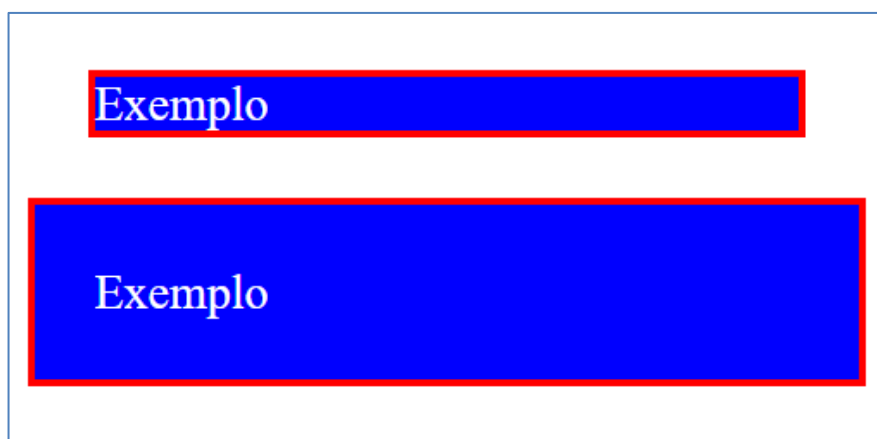
As propriedades border-color e border-style funcionam de forma semelhante à anterior com a identificação da cor ou estilo do limite ou limites. A propriedade border-radius permite associar um valor em “px” para arredondar os cantos dos limites.

PARTE VI – MARGIN E PADDING

Os espaçamentos de elementos são essenciais no trabalho com estilos, assim surgem os elementos margin e padding. O margin permite definir o espaço que vai desde o limite do elemento até ao elemento imediatamente colado (pode ser a própria página), enquanto que o padding aplica uma margem interna ao elemento, ou seja desde o limite até por exemplo ao texto que contém.

Vamos utilizar o exemplo de um parágrafo com limite vermelho, fundo azul e letras brancas, tudo num fundo branco:

```
.margin{  
  background-color:  blue;  
  border:2px solid  red;  
  color:  white;  
  margin: 20px;  
}  
  
.padding{  
  background-color:  blue;  
  border:2px solid  red;  
  color:  white;  
  padding: 20px;  
}
```



Na primeira linha conseguimos ver que a margem é externa ao elemento até à borda superior e bordas laterais da página e também ao elemento abaixo. Na segunda linha percebemos que as margens para as bordas são mínimas (padrão do HTML) e que a margem é interna ao elemento, neste caso entre a linha vermelha e o texto.

PARTE VII – WIDTH E HEIGHT

Este será um dos pontos mais importantes do CSS, a capacidade de redimensionar um elemento. Utilizando as propriedades width (largura) e height (altura) conseguimos colocar um qualquer elemento com o tamanho adequado. É possível definir cada atributo com os elementos:

- auto – valor por default. O browser calcula o tamanho;
- valor em px - permite definir o tamanho específico;
- % - define o tamanho com uma percentagem do bloco onde se insere.


Existem também propriedades para definição dos tamanhos máximo e mínimo de um elemento:

- max-height;
- max-width;
- min-height;
- min-width.

Exemplo:

```
<div class="cinzento">  
  <p class="metade">Exemplo</p>  
</div>
```

```
.cinzento{  
  background-color: grey;  
}  
.metade{  
  width: 50%;  
  background-color: lightblue;  
}
```



EXERCÍCIOS

1. Utilize a página de apresentação do CESAE criada na ficha anterior e aplique os seguintes elementos de CSS:
 - a. Toda a página terá uma cor de fundo “bisque” e margem de 20px;
 - b. As zonas header e footer devem ter o texto alinhado ao centro;
 - c. A imagem do header deve ter uma largura de 50% do elemento onde se encontra;
 - d. A cor de fundo da nav e do footer é grey; padding de 5px e margin de 10px;
 - e. A cor dos links da nav deve ser branca;
 - f. Os parágrafos devem ter o alinhamento justificado;
 - g. A tabela deve ter uma border de 1px tracejada e com lightblue; um padding de 10px e margin left e right como auto (centrar a tabela).

SABER MAIS – TRABALHO AUTÓNOMO

1. Altere o tipo de letra do texto para 'Lucida Sans' e a sua família.
2. Acrescente outros elementos de estilo à página. Pode ainda criar uma nova e complementar a informação.

Bom trabalho! 😊