



GUIA DE LABORATÓRIO 2.2 - TEXTO E EFEITOS

OBJECTIVOS

- Apresentar outros elementos HTML relacionados com conteúdo textual: BLOCKQUOTE, PRE, CODE, ARTICLE, etc.
- Utilizar outras propriedades relacionadas com estilização de texto: TEXT-DECORATION, TEXT-TANSFORM, FONT-VARIANT, etc.
- Algumas animações e transições com texto e com outros elementos

INSTRUÇÕES

PARTE I – PROPRIEDADES E CONSIDERAÇÕES RELACIONADAS COM TIPOS DE LETRA

1. Crie um novo ficheiro na pasta Lab2 com o nome Lab22Parte1.html .
2. Crie o esqueleto base de um ficheiro html, ou seja:

```
<!DOCTYPE html>
<html>
  <head>
  </head>
  <body>
  </body>
</html>
```

*Dentro de um elemento, o texto é colocado dentro da área de conteúdos desse elemento, começando no canto superior esquerdo (no caso de seguir ordem RTL - Right to Left) e "flui" de cima para baixo. O texto é um **elemento em linha**, como, por exemplo, âncoras (elemento **a**) ou imagens (elemento **img**). Quebras de linha ocorrem apenas quando a linha termina ou quando inserimos um elemento **br**.*

3. Vamos criar uma folha de estilos no próprio ficheiro html. Para tal devemos colocar a seguinte etiqueta no cabeçalho (**head**):

```
<!-- Seguem-se os estilos da folha de estilos -->
<style media="all" type="text/css">

</style>
```

4. Acrescente e alinhe devidamente o seguinte código:

```
<h1>Alberto Amaral</h1>

<p>Nasci em Agosto de 1971, em Almada, filho de <span>Armando Amaral</span>
    e de <span>Anabela Almeida</span>...</p>

<p>Fui a uma conferência sobre HTML e CSS. Encontrei o
    <span>Alberto</span> e o <span>Armando</span>. Convidei-os para jantar.
    Conversa mos sobre o <span>António</span>, que eles encontraram no
    casamento do <span>Arnaldo</span>... Umas semanas depois, fomos tomar um copo
    a casa do <span>Augusto</span>. Estava lá o <span>Adelino</span> que tinha
    chegado, com, surpresa!, o <span>António</span> e o <span>Arnaldo</span>.
    O <span>Arnaldo</span> já se tinha separado da esposa.
    Quando nos viu disse:
</p>

<blockquote cite="https://www.theknot.com/content/great-wedding-toastingquotes">
    <p>O casamento é uma grande instituição, mas eu não estou preparado
        para uma instituição.
    </p>
</blockquote>
```

*O elemento **blockquote** é, essencialmente, um elemento semântico de HTML. Elementos **semânticos** são elementos com significado, isto é, existem para descrever o conteúdo que contêm, independentemente do estilo que o navegador lhes aplica. Aliás, esse estilo é pouco importante porque, na maioria dos casos, é alterado via CSS. Exemplos de elementos não-semânticos: **div** e **span**. Exemplos de elementos semânticos: **article**, **form**, **table** e **blockquote**.*

*Note-se que neste caso teria sido melhor utilizar o elemento **em** do que o elemento **span**, uma vez que pretendemos realçar os nomes.*

5. Acrescente o seguinte código HTML:

```
<p>Ficámos a conversar sobre HTML e CSS, bem dispostos, até o dia nascer.
    Já agora, eis uma regra CSS:
</p>
<pre>
    div {
        border: 2px solid #f1d2e3;
        color: peru;
        background-color: yellowgreen;
        margin: 0;
    }
</pre>
```

*O texto dentro do elemento **pre** é exibido numa letra monoespaço e o navegador respeita todos os espaços e mudanças de linha. É o elemento ideal para ilustrar pedaços de código de linguagens de programação que queremos mostrar a um leitor da página.*

6. Queremos utilizar fontes disponíveis na Web, nomeadamente no serviço Google Fonts. Acrescente o seguinte elemento ao **head** da página:

```
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Tangerine">
```

7. A outra fonte será descarregada através da directiva (@-rule) **@import**. Acrescente no início da folha de estilos:

```
@import url('https://fonts.googleapis.com/css?family=Sunflower:300');  
@import url('https://fonts.googleapis.com/css?family=Roboto+Mono:400,700');
```

NOTA: Estas instruções apenas descarregam as fontes. Vamos incorporá-las nas regras que se seguem.

8. Comece por acrescentar as seguintes definições ao **body**:

```
. cor de fundo: rgb(230, 238, 248)  
. cor de texto: rgb(130, 143, 122)  
. fonte: Sunflower, Arial, Helvetica,  
       sans-serif  
. tamanho de letra: grande
```

9. Todos os parágrafos deverão possuir os seguintes estilos:

```
. cor de texto: rgb(126, 122, 0)  
. alinhamento de texto: justificado
```

10. Para o elemento **blockquote** acrescente as seguintes definições:

```
. cor de fundo: rgb(249, 249, 249)  
. estilo limite esquerdo: sólido  
. espessura limite esquerdo: 10px  
. cor do limite esquerdo: rgb(204, 204, 204)
```

11. Para todos parágrafos "filhos" de um elemento **blockquote** acrescente as seguintes definições:

```
. tipo de letra: Tangerine, cursive  
. tamanho: 28pt  
. cor: rgb(255, 145, 0)  
. o selector deverá ser blockquote > p
```

"Web Fonts" são fontes disponíveis em sítios Web e que são descarregadas pelo navegador antes de exibir o conteúdo das páginas. Uma possibilidade passa por utilizar a directiva CSS (@-rule) **@font-face** para indicar onde está alojada a fonte. Devemos incluir uma directiva **@font-face** por cada estilo (bold, italic, etc.). Esta directiva fica para um laboratório mais avançado, até porque temos alternativas mais simples, alternativas essas que manipulam a directiva **@font-face** por nós.

Podemos obter essas fontes de duas formas: elemento HTML **link** ou através de **@import**. Com o elemento **link**, referenciamos uma folha de estilos CSS externa, que contém as directivas **@font-face** todas para a fonte pretendida. Por seu turno, a directiva **@import** permite que uma folha de estilos inclua outras e é utilizada em outras situações, e não apenas para incluir Web Fonts.

Consultar:

https://www.w3schools.com/css/css3_fonts.asp

https://developers.google.com/fonts/docs/getting_started

<https://developer.mozilla.org/en-US/docs/Web/CSS/@import>

Pode especificar o limite com:

```
border-left: 10px solid rgb(204, 204, 204);
```

Também pode utilizar uma propriedade para cada característica:

```
border-left-width: 10px;  
border-left-style: solid;  
border-left-color: rgb(204, 204, 204);
```

Voltaremos a limites noutra laboratório mais avançado.

Nesta regra devemos utilizar um selector descendente directo para indicar que pretendemos que a regra seja aplicada a todos os **p** "filhos" de um **blockquote**. Um selector descendente corresponde a uma hierarquia de elementos sendo a regra aplicada ao elemento mais abaixo na hierarquia, isto é, ao elemento mais à direita do selector. Cada um dos elementos presentes no selector é separado por **>**.

Por exemplo, a seguinte regra ...

```
span > a {  
  color: red;  
}
```

... aplica-se a todos os **a** "filhos" de um **span**. Ao passo que a seguinte regra:

```
p > span > a {  
  color: red;  
}
```

... aplica-se a todos os **a** "filhos" de um **span** e "neto" de **p**.

Se na primeira regra separássemos o **a** do **span** apenas com espaços em vez de utilizarmos o **>**, ou seja, se a regra fosse ...

```
span a {  
  color: red;  
}
```

... então daríamos a cor vermelha a todas os **a** com um ascendente **span** ("pai", "avô", "bisavô", ...)

12. Após testar, acrescente um limite (em todas as direcções => utilize a propriedade **border**) ao **blockquote** idêntico ao do limite esquerdo, mas com espessura de **1px** apenas em vez de **10px**. Na regra para **blockquote**, esta propriedade deverá ficar **antes** da propriedade **border-left**. O que aconteceu?
13. Altere a regra para os parágrafos dentro de um **blockquote** de modo a que a margem externa seja **0** em todas as direcções.
14. Depois de testar, acrescente uma margem inferior de **0.5em**.
15. Acrescente uma curvatura aos cantos superior-direito e inferior-direito do limite (do **blockquote**):

```
border-radius: 0 5px 5px 0;
```

A propriedade **border-radius** permite definir a curvatura dos cantos dos limites:

```
border-radius: val; /* todos os cantos */  
border-radius: val1 val2; /* sup-esq + inf-dir, sup-dir + inf-esq */  
border-radius: val1 val2 val3; /* sup-esq, sup-dir + inf-esq, inf-dir */  
border-radius: val1 val2 val3 val4; /* sup-esq, sup-dir, inf-dir, inf-esq */
```

Consultar: https://www.w3schools.com/cssref/css3_pr_border-radius.asp

16. Queremos estilizar o bloco de citações colocando aspas personalizadas. Comece por acrescentar a seguinte regra:

```
blockquote p::before, blockquote p::after {  
    content: '""';  
    color: rgb(204, 204, 204);  
    font-size: 2.5em;  
    line-height: 0.5em;  
}
```

17. Remova a propriedade `content` da regra anterior e acrescente a seguinte propriedade à regra anterior:

```
quotes: '\201C' '\201D';
```

18. Crie as seguintes duas regras:

```
blockquote p::before {  
    content: open-quote;  
}  
blockquote p::after {  
    content: close-quote;  
}
```

19. Vamos estilizar os nomes e o bloco de código CSS. Acrescente uma regra para o elemento `span`:

```
span {  
    font-weight: bolder;  
    font-variant: small-caps;  
}  
pre {  
    font-family: 'Roboto Mono', Courier, 'Courier New', monospace;  
    font-size: 12pt;  
}
```

20. Vamos estilizar a primeira linha do elemento `pre` com o pseudo-elemento `::first-line`:

```
pre::first-line {  
    font-weight: bold;  
}
```

Os selectores `::before`/`::after` criam um **pseudo-elemento** que é o primeiro/último **descendente** do elemento seleccionado. Tipicamente, é utilizado para ornamentar um elemento através da propriedade `content`. O elemento adicionado é um elemento **em linha**.

Note que, neste caso, combinámos um selector `::before`/`::after` com um selector descendente para indicar que queremos adicionar conteúdo a todos os parágrafos dentro de um `blockquote`.

Consultar:

<https://developer.mozilla.org/en-US/docs/Web/CSS/::before>

A propriedade `content` permite adicionar conteúdo a um elemento e é utilizada com os selectores `::before`/`::after`.

O conteúdo mais simples é um pedaço de texto, mas temos outras possibilidades. Vamos utilizar os valores `open-quote` e `close-quote` para indicar que queremos aspa de abertura e de fecho definidas pela propriedade `quotes`.

Consultar:

<https://developer.mozilla.org/en-US/docs/Web/CSS/content>
https://www.w3schools.com/cssref/pr_gen_quotes.asp
https://www.w3schools.com/cssref/css_entities.asp

Também tem à disposição o pseudo-elemento `::first-letter` para estilizar o primeiro caractere.

21. A partir da Google Fonts, obtenha a fonte Ubuntu Mono. Obtenha a versão regular/normal e negrito desta fonte (encontra ajuda no Google Fonts). Altere a folha de estilos de modo a que passe a ser esta a fonte do elemento **pre**.

22. Acrescente as seguintes propriedades ao atributo style do terceiro parágrafo ("Ficámos a conversar..."):

```
text-decoration: line-through; text-transform: capitalize;
```

23. Modifique o valor da propriedade para **text-decoration** para underline overline wavy (sim, pode especificar mais do que um valor).

24. Acrescente o seguinte código HTML:

```
<h2>lorem ipsum dolor</h2>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam congue erat
    ut mauris pharetra, ac tempor lectus facilisis. Mauris id ipsum nisl.
    Vivamus nec erat odio. Morbi vitae orci id tortor interdum placerat.
    Nulla augue enim, tincidunt in ipsum in, scelerisque ullamcorper velit.
</p>
```

25. Estilize o h2 com:

- . estilo de fonte: itálico
- . transformação de texto: capitalizado
- . font-variant: "maiúsculas pequenas"

26. Vamos dar um efeito ao texto com a propriedade **text-shadow**. Comece por acrescentar:

```
text-shadow: 0px 0px 1px rgb(130, 143, 122);
```

27. Agora mude o primeiro valor para 30px, teste, altere o segundo para 30px, e teste novamente. O que conclui?

28. Altere o terceiro valor para 5px e depois para 10px.

29. Modifique os primeiros dois valores para -2px e o último para 5px.

Exemplo de utilização de **text-shadow**:

```
/*offset-x offset-y blur-radius color */
text-shadow: 1px 1px 2px black;
```

Pode obter uma descrição da propriedade **text-shadow** através da ligação seguinte:

<https://developer.mozilla.org/en-US/docs/Web/CSS/text-shadow>

30. Acrescente agora a seguinte regra:

```
h2 + p {  
    text-align: left;  
    margin-left: 2em;  
    margin-right: 2em;  
    text-align-last: center;  
}
```

O selector `elemento1 + elemento2` permite estilizar todas as ocorrências de `elemento2` que são precedidas por `elemento1`.

A propriedade `text-align-last` indica como a última linha de um bloco, ou uma linha, mesmo antes de uma quebra de linha forçada via `br`, deve ser alinhada.

31. Finalmente, investigue a directiva `@keyframes`, bem como as propriedades `animation` e `animation-duration`. Acrescente o seguinte código:

```
@keyframes dir_para_esq {  
    from {  
        margin-left: 100%;  
        width: 300%;  
    }  
    to {  
        margin-left: 2em;  
        width: 100%;  
    }  
}
```

32. Acrescente as seguintes propriedades à regra `h2 + p`, teste e explique o sucedido:

```
animation-name: dir_para_esq;  
animation-duration: 1s;
```