

# Introdução às Tecnologias Web

## Guião da Aula Teórico-Prática sobre jQuery (Parte 2)

### Antes da aula

No tutorial de jQuery do W3Schools, ler as secções de HTML Get/Set/Add, Hide/Show Effects, bem como o guia de referência de jQuery Events.

### Durante a aula

Comece por obter a sua resolução dos exercícios da aula anterior, sobre **músicas e tops semanais**, assumindo-se que, pelo menos, a lista de todas as músicas está a ser bem preenchida.

Os objetivos desta aula são fazer a conversão de código JavaScript existente para jQuery, a fim de constatar se fica ou não mais simples, praticar a manipulação do DOM (*Document Object Model*), incluindo o acesso a um campo de formulário, e treinar o uso de uma animação.

**Exercício 1:** Relembre os elementos e atributos em `index.html`, e as constantes, variáveis, e funções em `comportamento.js`. Analise que código é executado quando o *browser* termina o carregamento do documento HTML e quais as funções em que não está a ser usada a biblioteca jQuery, podendo sê-lo. Por fim, no *browser*, inspecione a estrutura HTML da lista de todas as músicas, integralmente gerada por código JavaScript.

**Exercício 2:** No ficheiro `comportamento.js`, converta o código de `desenhaListaMusicas()` para jQuery. Comece por mudar o nome do parâmetro `elementoPai` para `$elementoPai`.<sup>1</sup> O prefixo `$` indica que se espera que os argumentos passados à função através desse parâmetro sejam objetos jQuery. Consequentemente, na última instrução, altere `appendChild()` para `append()`, sendo este o correspondente método jQuery. Depois, reescreva o restante código da função e verifique se ficou mais simples de interpretar. Não experimente ainda recarregar a página `index.html`, pois primeiro é preciso fazer alterações noutras partes do código.

**Exercício 3:** Converta o código da função `desenhaTodasMusicas()` para jQuery. Certifique-se que o primeiro argumento passado a `desenhaListaMusicas()`, atualizada no exercício anterior, é um objeto jQuery. Agora já pode experimentar carregar novamente a página `index.html` e deverá ver a lista de todas as músicas, mas não as músicas dos *tops* semanais...

**Exercício 4:** Atualize a função `desenhaTopsSemanais()` para passar a usar as funcionalidades jQuery. Tal como no exercício anterior, tenha especial cuidado com o primeiro argumento passado a `desenhaListaMusicas()`.

---

<sup>1</sup> No Visual Studio Code, para mudar o nome de um parâmetro, variável, constante, ou função, deve colocar o cursor em cima do nome e premir a tecla F2 (*Rename Symbol*). O novo nome irá substituir o antigo dentro do respetivo âmbito de visibilidade. Isto significa, por exemplo, que mudar o nome de uma variável declarada dentro de uma função não vai afetar uma variável homónima noutra função.

O propósito dos próximos exercícios é desenvolver um filtro sobre a lista de todas as músicas, controlado a partir de um campo de texto. Assim que o utilizador escrever algo nesse campo, as músicas da lista que não contiverem esse algo devem desaparecer.

**Exercício 5:** Em `index.html`, acrescente um formulário com um campo de texto antes da lista de todas as músicas. Guarde o identificador do campo de texto numa constante chamada `CAMPO_FILTRO_MUSICAS`. Note que não é preciso declarar uma constante com o identificador do formulário, como é conveniente fazer em JavaScript, pois em jQuery os dados de quaisquer campos de formulários podem ser diretamente acedidos pelos respetivos identificadores.

**Exercício 6:** Em `comportamento.js`, crie a função `trataFiltrarMusicas()`, sem parâmetros, que acede ao campo do filtro e, para cada item da lista de todas as músicas (não abrange as listas dos *tops* semanais), verifica se o texto do filtro faz parte do título ou artista da música e, em caso negativo/positivo, faz uso de uma animação lenta para esconder/mostrar esse item da lista. Pode ser útil recorrer a `String.toLowerCase()`, para ser irrelevante o uso de maiúsculas e minúsculas nos dados da música e no texto do filtro, `String.includes()`, bem como às funções da biblioteca jQuery `$.each()`, `$.show()`, e `$.hide()`. No *browser*, experimente escrever algo no campo do filtro e, na consola, invoque a função para ver se só aparecem as músicas correspondentes.<sup>2</sup>

**Exercício 7:** Na função `principal()`,<sup>3</sup> defina `trataFiltrarMusicas()` como sendo um *event listener* a chamar sempre que haja alterações no campo de texto do filtro. A lista de todas as músicas deve ser atualizada de imediato, sem necessidade do utilizador carregar na tecla Enter. Faça testes com os eventos `keyup` e `keypress` e decida qual dos dois é o mais adequado.

---

<sup>2</sup> Em vez de `$.show()`, e `$.hide()`, pode experimentar usar `$.fadeTo()` para desvanecer parcialmente (isto é, sem fazer desaparecer) as músicas que não contiverem o texto no filtro. Neste caso, os itens da lista são sempre os mesmos, sendo mais fácil perceber os efeitos do filtro.

<sup>3</sup> Alternativamente, as definições de *event handlers* podem ser movidas para uma função própria, por exemplo com nome `defineEventHandlersParaElementosHTML()`, como foi feito numa aula teórico-prática anterior.