

# MÓDULO

# JAVASCRIPT/AJAX

PRÁTICA

APLICAÇÃO DE JQUERY COMO ELEMENTO  
DINÂMICO DE CONTEÚDO NAS PÁGINAS DA  
WEB

## APLICAÇÃO DE JQUERY COMO ELEMENTO DINÂMICO DE CONTEÚDO NAS PÁGINAS DA WEB

---

### 1.1. DESCRIÇÃO DA PRÁTICA

**Tempo estimado:** 2 h.

REQUISITOS
<ul style="list-style-type: none"><li>■ Ter concluído a unidade didática “Conceitos básicos de AJAX e JSON e jQuery”.</li></ul>
OBJETIVOS
<ul style="list-style-type: none"><li>■ Compreender o uso de jQuery, uma biblioteca JavaScript.</li><li>■ Aprender o uso de métodos e funções jQuery para criar efeitos da web.</li></ul>
MATERIAL E FERRAMENTAS NECESSÁRIAS PARA A PRÁTICA
<ul style="list-style-type: none"><li>■ Editor de código.</li></ul>

## 1.2. DESENVOLVIMENTO DA PRÁTICA

Nesta prática, verá o uso de jQuery como um elemento dinâmico de conteúdo nas páginas da web.

### Uso de bibliotecas alojadas num servidor externo

---

Para utilizar as funcionalidades do jQuery sem ter a biblioteca alojada localmente, terá apenas de obter a URL do ficheiro .js necessário. Neste exemplo, é utilizada a biblioteca jQuery do servidor do Google para hospedar recursos de desenvolvimento da web:

```
<script  
src="https://ajax.googleapis.com/ajax/libs/jquery/1.6.1/jquery.min.  
js"> </script>
```

É importante referir que nunca se deve carregar um ficheiro JavaScript externo se ele não for de uma fonte confiável.

Embora possa ter uma infinidade de aplicações que utilizem jQuery no código, um dos mais úteis é o uso das funções ou métodos `fadeIn()`, `fadeOut()` e `fadeTo()`. Estas funções alteram os valores de visibilidade das divs escolhidas, podendo fazer alterar a visualização dos elementos da página web. Assim, o `fadeIn()` altera a visibilidade de oculto para visível, o `fadeOut()` altera a visibilidade de visível para oculto e o `fadeTo()` altera a percentagem de visibilidade de um elemento.

Nesta prática verá o seu uso como uma forma introdutória e as possibilidades que estas funções oferecem.

Para começar, deverá utilizar os ficheiros fornecidos na pasta de apoio a esta prática disponibilizada no seu Campus Virtual. Os ficheiros HTML e CSS foram já desenvolvidos e as imagens utilizadas também. O resultado da visualização será:

## Fading em jQuery

**Mostrar e ocultar elementos de forma suave com "fading methods"**

Ocultar toda a caixa

Ocultar caixa

Ocultar a lista abaixo

Ocultar

Mostrar a lista abaixo

Mostrar

Ocultar e mostrar a lista abaixo

Ocultar/Mostrar

Escolha a percentagem de fading da lista: 100% ▾

Escolhe um valor e um elemento para ativar o seu fading:



0

e1 ▾

Alterar

Reset

- Elemento 1
- Elemento 2
- Elemento 3

Começemos por analisar o cabeçalho <head> do documento HTML:

```
<link href="estilos.css" rel="stylesheet" type="text/css">

<script
src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.
js"></script>

<script src="scriptjquery.js"></script>
```

Como pode ver, foram declarados três ficheiros diferentes.

- O primeiro, o ficheiro de estilo CSS, que dará estilo à página web.
- O segundo, a biblioteca jQuery hospedada nos repositórios de código do Google.
- O terceiro, o ficheiro JavaScript criado à parte nesta prática, com o código jQuery das funções que serão analisadas aqui.

Agora vamos começar a trabalhar no arquivo .js. Iremos abrir o editor de código e iniciar a programação jQuery. Nesta prática específica, executaremos as nossas funções assim que o documento for carregado com a seguinte frase:

```
$(document).ready(function(){ }
```

Esta linha significa que, quando o documento estiver carregado e pronto, as seguintes funções são executadas quando necessário. A partir desta frase, que abrangerá todo o script e o seu conteúdo, poderá interagir com os restantes elementos do ficheiro HTML.

No primeiro exemplo que verá, é identificada a div principal que é vista na imagem (caixa1), e é ocultada e exibida novamente um segundo depois. Os códigos HTML e jQuery seriam:

HTML:

```
<div class="caixa1" id="caixa1">
```

jQuery:

```
$("#ocultarCaixa").click(function(e){  
    $("#caixa1").fadeOut(function(){  
        $(this).delay(1000);  
        $(this).fadeIn();  
    });  
});
```

Tudo o que a div com o id “caixa1” contém será oculto pela função principal #ocultarCaixa, que irá ser chamado por um botão mostrado a seguir; este estará dentro da div principal “caixa1”:

```
<button class="botao cinza" type="button" id="ocultarCaixa">Ocultar  
Caixa</button>
```

Ao clicar neste botão, este irá chamar a função feita anteriormente, ou seja, ele irá fazer com que a div “caixa1” desapareça com o fadeOut, que tenha um atraso de um segundo e, por fim, que a div volte com o fadeIn.

Agora, pretende-se ocultar apenas um item, neste caso de uma lista não ordenada com três itens.

```
<ul id="lista">  
  
    <li id="e1"><h2>Elemento 1</h2></li>  
  
    <li id="e2"><h2>Elemento 2</h2></li>  
  
    <li id="e3"><h2>Elemento 3</h2></li>  
  
</ul>
```

Neste excerto de código, é criada uma lista não ordenada com o id “lista”, e cada elemento dessa lista também tem o seu próprio id.

```
<button class="botao cinza" type="button"
id="ocultarTudo">Ocultar</button>
```

Depois de criar o botão responsável por chamar a função que fará desaparecer toda a lista, irá então fazer essa mesma função:

```
$("#ocultarTudo").click(function(e){
    $("#lista").fadeOut();
});
```

Para fazer a lista voltar a aparecer, terá de fazer uma função idêntica, mas desta vez utilizando o fadeIn() para exibir novamente o conteúdo da lista.

HTML:

```
<button class="botao cinza" type="button"
id="mostrarTudo">Mostrar</button>
```

jQuery:

```
$("#mostrarTudo").click(function(e){
    $("#lista").fadeIn();
});
```

O exemplo a seguir é uma mistura dos dois anteriores, praticamente idêntico à div "caixa1", mas com os elementos da lista acabadas de utilizar.

HTML:

```
<button class="botao cinza" type="button"
id="ocultarMostrar">Ocultar/Mostrar </button>
```

jQuery:

```
$("#ocultarMostrar").click(function(e){  
    $("#lista").fadeOut(function(){  
        $(this).delay(1000);  
        $(this).fadeIn();  
    });  
});
```

À semelhança da função utilizada na div “caixa1”, é invocado o método `fadeOut()` ao clicar no botão, espera um segundo e invoca o método `fadeIn()` no seletor “#lista”.

O próximo ponto nesta prática é trabalhar com outro tipo de elemento para controlar o desaparecimento dos elementos na página web. Neste caso, será trabalhado com um seletor com quatro valores, que será lido pelo jQuery para atuar com o método `fadeTo()`.

A primeira coisa a criar é a lista de seleção a partir da qual poderá escolher a percentagem de opacidade dos elementos da lista já criada e atribuir um id.

```
<select name="opacidade" id="selOpacidade">  
    <option value="0.2">20%</option>  
    <option value="0.5">50%</option>  
    <option value="0.8">80%</option>  
    <option value="1" selected="1">100%</option>  
</select>
```

Com a lista criada, e com uma seleção padrão de 1 (o valor mais alto das opções disponíveis), falta agora programar a função de mudança de opacidade.

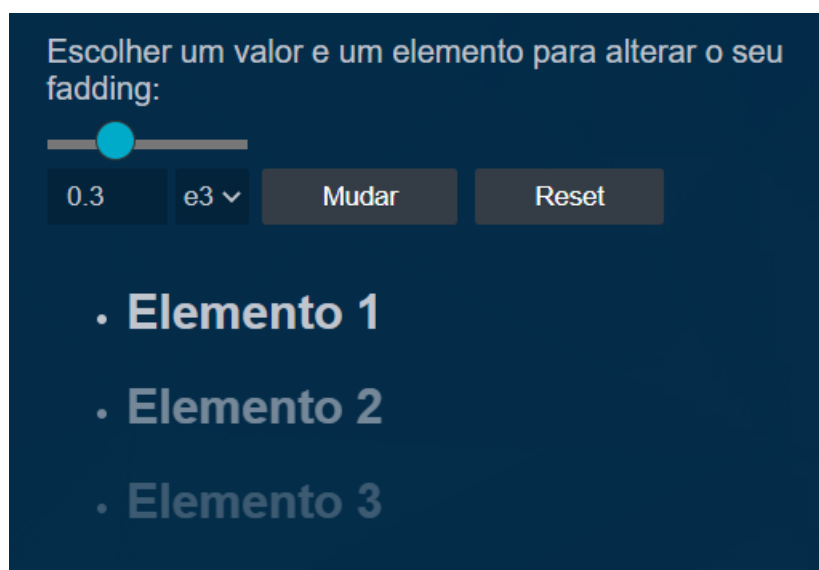


```
$("#selOpacidade").change(function(e){
    var opacidadeEsc = e.target.options[e.target.selectedIndex].value;
    $("h2").fadeTo(500, opacidadeEsc);
});
```

Nesse caso, foi utilizado o `change()` em vez de `click()`, pois pretende-se controlar a alteração do valor na lista selecionável. Ao selecionar um valor diferente do atual, a percentagem de opacidade mudará automaticamente com base no valor escolhido. Isso é possível através da variável "opacidadeEsc" que é exatamente o índice selecionado (`selectedIndex`). Foi o elemento `h2` em vez da "#lista", pois é o estilo de título atribuído à lista. Tem um lado negativo, que é alterar a opacidade de todos os `h2` na página, embora, para este exemplo, seja perfeitamente útil.

Dentro das opções do método `fadeTo()`, o número 500 é o tempo que dura a transição (em milissegundos) e a variável "opacidadeEsc" indica o valor selecionado na lista.

O último elemento de controlo da opacidade dos itens da lista irá mudar individualmente para cada elemento, e será uma entrada do tipo `range`.



O HTML para o criar é o seguinte:

```
<input type="range" class="slider" id="percentagem"
name="percentagem" max="1" step="0.1" value="0">
```

Também é necessário criar uma caixa de texto para exibir o valor selecionado:

```
<input class="textbox" type="text" name="porcento" id="porcento"
value="0" maxlength="3" size="3" disabled>
```

A função jQuery faz com que o valor do seletor passe para a caixa de texto (que está disabled para que não possa ser modificado pelo utilizador):

```
$("#percentagem").change(function(){
    $("#porcento").val($(this).val());
});
```

Nesta função, cada vez que o seletor muda de valor, é dito à caixa de texto para exibir esse mesmo valor automaticamente, com um valor inicial de zero.

Segue-se a função jQuery para modificar a visibilidade dos elementos da lista com base no que for escolhido "selElemento".

```
$("#mudar").click(function(e){
    var opacidade = $("#porcento").val();
    var elemento = "#" + $("#selElemento option:selected").html();
    $(elemento).fadeTo(500, opacidade);
});
```

Pode verificar que, ao clicar no elemento "#mudar" (que é um botão), é iniciada a função que armazena o valor da opacidade e a opção selecionada, dentro de duas variáveis. De seguida, atua sobre o elemento escolhido (ou melhor, o valor da variável "elemento"), mudando a sua opacidade para o valor da variável "opacidade" utilizando o método fadeTo().

Por último, mas não menos importante, um botão que recarrega a página web para todos os valores voltarem ao seu estado inicial, pois, depois de testar um pouco os intervalos, os fades e os vários botões, no final a opção mais rápida para voltar tudo ao seu estado inicial é:

```
$("#resetFade").click(function(e){  
    location.reload();  
});
```

Com este pequeno código terá a opção de reset a todos os valores modificados.

