Ficha Prática nº 10 – *Introdução ao Javascript*

Nota: Não deve alterar o documento HTML nem o ficheiro estilos.css.

Exercício 1: Introdução ao *Javascript – alert()*

1. Efetue o download e descompacte o ficheiro *ficha10.zip*.
2. Visualize a página ficha10.html no browser (Figura 1).

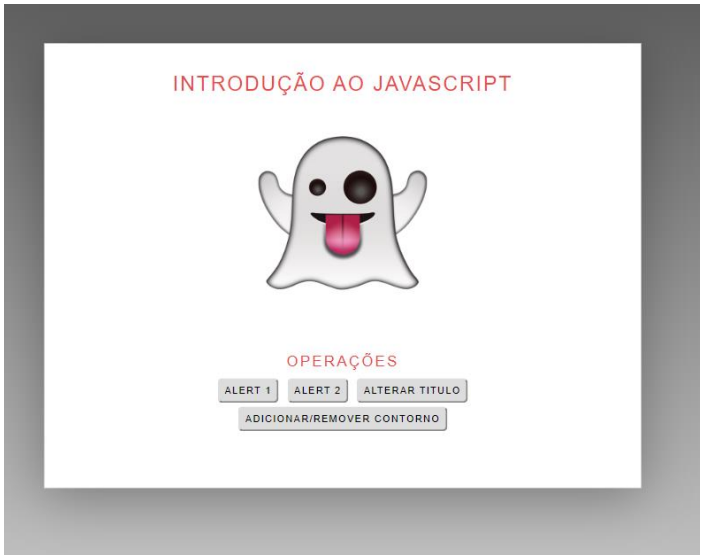

Figura 1 - Alert do Exercício 1)

3. Analise o código do botão **Alert 1** e verifique que este inclui um *script* em *Javascript*, embebido em HTML e controlado por um evento (*onclick*), que permite apresentar a mensagem **"Exercício Javascript"** sempre que se clica no botão.

   A apresentação da mensagem é efetuada através do método ***alert()*** como demonstra o código seguinte:

```
<button onclick="alert('Exercício Javascript')">ALERT 1</button>
```

Figura 2 - Alert do Exercício 1)

4. Verifique que o comportamento anterior pode ser associado a outros eventos, como por exemplo, ***onmouseover*** (sobreposição do cursor do rato no botão), ***onmouseout*** (eliminação da sobreposição do cursor do rato no botão). Substitua o evento e visualize o comportamento obtido na janela do *browser*. Reponha o valor inicial.

5. De forma a existir uma separação clara entre os elementos de estrutura (HTML) e os respetivos comportamentos (JavaScript), é possível assegurar o funcionamento da alínea anterior efetuando toda a implementação exclusivamente no *script*. Desta forma, **não existirá** qualquer código *JavaScript* diretamente integrado em código HTML.

   Para tal recorre-se a um *event listener* implementado através do método addEventListener() (anexo I). Assim, tendo por base o *addEventListener(),* implemente um *event listener* para detetar um *click* no botão **Alert 2***,*:

   1. declarando o processamento numa função externa;
   2. recorrendo a uma *anonymous function* na qual se efetua a chamada à função anterior (externa);
   3. recorrendo a uma *anonymous function* na qual se implementa diretamente o processamento pretendido.

6. Visualize no browser e certifique-se que sempre que existe um *click* no botão "Alert 2" é apresentada a mensagem correta. Note que não foi especificado qualquer código JavaScript no código HTML.

   Caso não esteja a obter o resultado pretendido, verifique se existe algum erro de código. Para tal aceda às *Developer Tools (ex: Chrome F12)* do *browser* e na janela apresentada selecione o *tab* "Console". Será nesta zona onde serão apresentados os erros de *JavaScript*.

Exercício 2: Botão "Alterar Titulo"

1. Pretende-se que o botão **Alterar Título** origine três comportamentos distintos:

   a. altere o texto do título quando se clica (*click*) (Figura 3);

   b. enquadre o texto do título num elemento <h4> sempre que se verifique uma situação de sobreposição do cursor do rato (*mouseover*) com o botão (Figura 4);

   c. reponha o estado inicial quando se elimina a sobreposição do cursor com o botão (*mouseout*) (Figura 5).

Figura 3 – Click no botão

Figura 4 - Rato em cima do botão, incluído num <h4>

Figura 5 - Estado normal (sem rato em cima do botão)

Adicione os três *event listener* necessários para implementar o comportamento descrito.

Nota: Deve utilizar as propriedades *textContent* e *innerHTML* assim como explorar o método *querySelector()*.

2. Visualize a página no browser e verifique se o comportamento é o desejado. Recorra às *Developer Tools (Console)* para identificação dos possíveis erros existentes.

Exercício 3: Botão "Adicionar/Remover Contorno"

1. Pretende-se que o botão "Adicionar/Remover Contorno" aplique a classe **border-active** (já definida no ficheiro CSS) e cor de fundo #FF000022 ao elemento **panel-animals**, ou remova estes estilos, caso estes já estejam aplicados. As figuras seguintes demonstram o comportamento desejado:

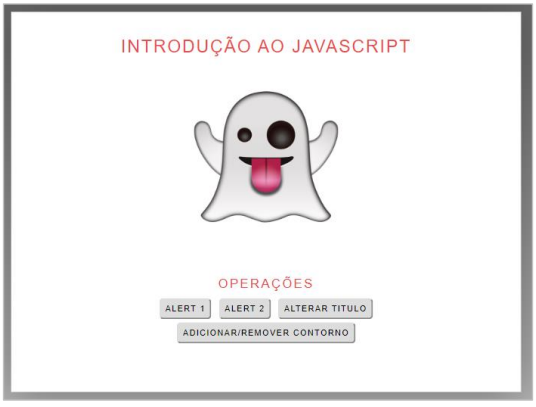

Figura 6 - Remoção do Contorno

Figura 7 - Adição do contorno

2. Adicione um *event listener* para implementar o comportamento deste botão, sabendo que se pretende que este comportamento seja alterado através de um clique no botão.
3. Implemente o comportamento pretendido: i) testa se a *class* contém o valor *border-active*; ii) se sim elimina esse valor do atributo *class*; iii) caso contrário adiciona essa *class*.

Nota: Deve utilizar a propriedade *classList*, assim como os métodos *contains(); add(); remove().*

4. Altere a cor de fundo do elemento **panel-animals** tendo por base o método *setAttribute()***.**
5. Efetue a mesma alteração mas desta vez recorrendo à propriedade *style*. Deve comentar o código implementado na alínea anterior.
6. Declare uma variável de forma a eliminar a necessidade de utilizar repetidamente o método *querySelector()* para referenciar o elemento cuja *class* é *panel-animals*.

Anexo I

a. O código javascript pode ser incluído num documento HTML de forma embebida, através da tag <script>, ou então com recurso a um ficheiro externo *.js especificado como valor do atributo src da tag <script>.

b. A sintaxe genérica para definir um *event listener* é a seguinte:

```
elemento.addEventListener(event, function, useCapture);
```

- ***elemento*** – Elemento ao está associado o evento;
- ***event*** – Qual o evento a capturar (ex: click);
- ***function*** – Qual a função a executar
- ***useCapture*** – Parâmetro opcional que indica se deve haver encadeamento de eventos

c. Existem várias formas de se especificar um elemento ao qual se pretende adicionar um *event listener*. Visualize alguns exemplos:

`document.querySelector('#btn-titulo')`	Permite obter o elemento cujo id é btn-titulo
`document.querySelector('.btn-border')`	Permite obter o 1º elemento com a classe btn-border
`document.getElementById('btn-reset')`	Permite obter o elemento cujo id é btn-reset

d. Três formas distintas de aplicar o método addEventListener():

1. Declarando uma função externa e especificando como argumento o nome dessa função. Esta função pode ser invocada por outros *event listeners* ou outras funções.

```
function exemplo() {
    alert("Exercicio Javascript - Alert 2")
}
elemento.addEventListener('click', exemplo);
```

Nota Importante: o nome da função externa é especificado sem parêntesis.

2. Recorrendo, a uma função sem nome (*anonymous function*) na qual se efetua uma chamada à função externa:

```
function exemplo() {
    alert("Exercicio Javascript - Alert 2")
}

elemento.addEventListener('click', function () {
    exemplo();
});
```

3. Recorrendo, a uma função sem nome (*anonymous function*) na qual implementa o processamento pretendido.