

## > Ficha Prática Nº2 (JavaScript – Variáveis, Event Listeners, Funções)

### Notas:

- Os alunos **não devem alterar** o documento HTML nem os ficheiros de estilos existentes, de forma a seguirem o propósito da ficha.
- Não devem remover** a instrução `'use strict'` que se encontra no topo do ficheiro `index.js` de forma a que seja usada na implementação, uma variante mais restrita do *JavaScript*.
- Pretende-se, nas próximas aulas, implementar, em JavaScript, o tradicional jogo de memória que terá o aspeto apresentado das figuras seguintes. Recomenda-se, para cada ficha, ler as dicas (apresentadas na seção **Dicas**) antes de iniciar resolução da ficha. O HTML, bem como o CSS necessário à resolução, já inclui todos os elementos necessários.



Figura 1 – Jogo de Memória em JavaScript – Imagens da aplicação

## > Dicas para resolução da ficha:

- a. Concentre a declaração das variáveis globais no topo do **index.js**, depois da instrução **'use strict';** por forma a facilitar a sua localização.
- b. Variáveis **que não serão alteradas**, devem ser declaradas com **const**, caso contrário, declare com **let**. Recomenda-se não utilização da declaração recorrendo ao **var**.
- c. Existem várias formas de se aceder a um elemento DOM, seja para alterar o seu estado ou adicionar um *event listener*. Exemplos:

<code>document.querySelector('#elemento')</code>	> Permite obter o elemento cujo id é <b>elemento</b>
<code>document.querySelector('.elemento')</code>	> Permite obter o 1º elemento com a classe <b>elemento</b>
<code>document.getElementById('elemento')</code>	> Permite obter o elemento cujo id é <b>elemento</b>
<code>document.querySelectorAll('.elemento')</code>	> Permite obter <b>todos</b> os elementos com a classe <b>elemento</b>

- a. A **alteração de estilos** de um elemento pode ser efetuada com recurso à propriedade *style*, da seguinte forma:

```
elemento.style.display = 'none';
```

- b. A **alteração do texto** de um elemento deve ser efetuada com recurso à propriedade :

```
elemento.textContent = 'JavaScript';
```

- c. A **propriedade classList** permite obter o nome das classes de um elemento. É possível adicionar, remover ou alternar entre duas classes através dos métodos `add()`, `remove()` e `toggle()`, respetivamente.

```
elemento.classList.add("estilo");
```

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

```
elemento.addEventListener(e,function,useCapture)
```

- > **elemento** – Elemento que se está a associar o evento;
- > **e** – Evento a capturar (ex: click)
- > **function** – Função a ser executada
- > **useCapture** – Parâmetro opcional que indica se deve haver encadeamento de eventos

O método `addEventListener` permite anexar um *event handler* a um determinado elemento.

e. Existem várias formas de aplicar o método **addEventListener()**, como apresentado nos exemplos abaixo:

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

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

```
const elemento = document.querySelector('h1');
elemento.addEventListener('click', exemplo);

function exemplo() {
    console.log('O elemento h1 foi clicado!');
}

// ou então
function exemplo(event) {
    console.log(`O elemento ${event.target.tagName} foi clicado!`);
}
```

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

```
const elemento = document.querySelector('h1');
function exemplo(msg) {
    console.log(msg);
}
elemento.addEventListener('click', function () {
    exemplo('Elemento Clicado! Funcao com parametros!');
});

// outra forma...
elemento.addEventListener('click', () =>
    exemplo('Elemento Clicado! Funcao com parametros!'));
```

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

```
const elemento = document.querySelector('h1');

elemento.addEventListener('click', function () {
    console.log('O elemento foi clicado!');
});

// ou então com arrow function
elemento.addEventListener('click', () => {
    console.log('O elemento foi clicado!')
});
```

**f.** A sintaxe genérica para definir um **for..of** é a seguinte:

```
for (variavel of iteravel) {
    //... código ser executado
}
```

**g.** A sintaxe genérica para definir um **forEach** é a seguinte:

```
elementos.forEach(function(elemento, index, arr)) {
    //...
});
```

- > **function** – função a ser executada por cada elemento
- > **index** – opcional, índice do elemento corrente
- > **arr** – opcional, array do elemento corrente

### > Preparação do ambiente – Efetue os seguintes passos:

- a. Efetue o download e descompacte o ficheiro **ficha2.zip** disponível no *inforestudante*.
- b. Inicie o *Visual Studio Code*, abra a pasta no **workspace** e visualize a página **index.html** no browser, no qual terá o aspeto da figura 2.
- c. Como pode verificar, a mesma não apresenta o aspeto necessário do início do jogo, e será esse o propósito da ficha, escondendo/mostrando alguns elementos iniciais, com recurso a *event listeners*.

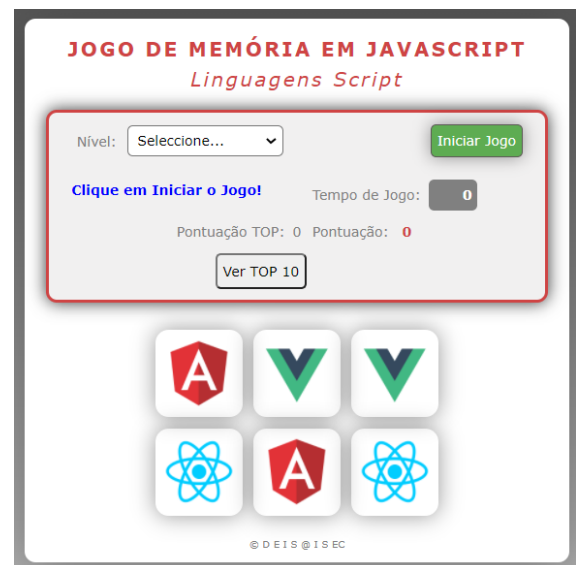


Figura 2 - Jogo (início)

## Parte I – Implementação função Reset

- 1>** Nesta fase, pretende-se criar uma função que vai especificar o aspeto inicial do jogo. A figura 3 apresenta o aspeto pretendido. Para facilitar a implementação, siga os passos e consulte as dicas anteriormente apresentadas.

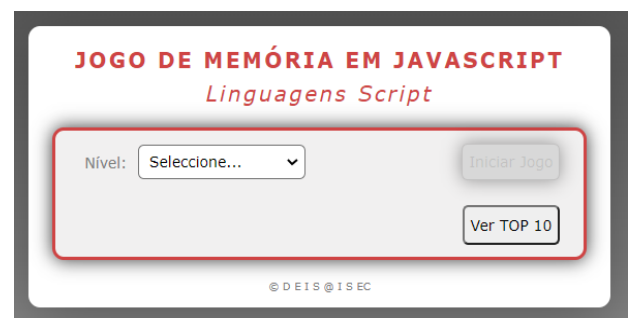


Figura 3 – Início do Jogo

a. Especifique, no ficheiro **index.js**, as seguintes **variáveis** contantes, logo após a declaração **'use strict'** existente no ficheiro:

- **message** que permite aceder ao elemento **#message**
- **panelControl** que permite aceder ao elemento **#panel-control**
- **panelGame** que permite aceder ao elemento **#panel-game**
- **btLevel** que permite aceder ao elemento **#btLevel**
- **btPlay** que permite aceder ao elemento **#btPlay**

*Nota: Ver Dicas*

b. Implemente a função **reset()** no ficheiro **index.js** de forma a completar os seguintes passos:

1. Deverá esconder o panelGame, especificando propriedade **display** com **none**;
2. Invoque a função **reset**, através da respetiva chamada **reset()** e confirme o resultado no *browser*, certificando-se que não existe nenhum erro de código JavaScript.
3. Elimine o texto existente no elemento **message**, especificando a propriedade **textContent=""**, e **coloque o elemento visível**, removendo a class **hide**, com recurso à propriedade **classList** e método **remove**. `message.classList.remove('hide');`
4. Crie uma variável que obtenha todos os elementos especificados com a classe **.gameStarted** existentes no painelControl. Deve usar o **querySelectorAll**.
5. Com recurso ao **for... of** ou com o **forEach**, percorra todos os elementos obtidos na alínea anterior de forma a adicionar a class **'hide'**. Deve usar a propriedade **classList** e o método **add**.
6. Implemente o código de forma a que o elemento **#btPlay** fique desativado. A propriedade que permite esse comportamento é **disabled = true**

c. Confirme, no *browser*, se o resultado final tem o aspeto da figura 3.

## Parte II – Implementação de Event Listeners

2> Nesta segunda parte, pretende-se que quando se seleciona/clica em alguns elementos, aconteça algum comportamento. Assim, sempre que se selecionar um nível (btLevel), o botão de iniciar jogo (btPlay) fica ativo, caso contrário, fica desativo, como mostram as figuras seguintes.



Figura 4 - Opção '0' selecionada



Figura 5 - Opção '1' selecionada

- a. Para implementar este comportamento, altere a declaração implementada na função **Reset** no qual o botão “Iniciar Jogo” (**#btPlay**) fica sempre desativo, para **ficar apenas desativo** se a opção selecionada for “Selecione...”, opção cujo valor é ‘0’ (podem confirmar no html), caso contrário, se for selecionado um dos níveis, o elemento **#btPlay** **deve ficar activo**.

Para obter o valor da opção selecionada no elemento *select*, deve recorrer à propriedade **value**.

- b. Também o **panelGame** deverá ficar visível quando se seleciona um nível, aplicando para isso o estilo **grid** à propriedade **display**.
- c. Implemente um *Event Listener* de forma a que, sempre que houver uma alteração à opção selecionada (evento **change**) no elemento **btLevel**, seja executada esta função **reset**.
- d. Confirme, no *browser*, se o resultado final tem o comportamento pretendido.

- 3> Tendo em conta tudo o que implementou nas alíneas anteriores, implemente o código necessário de forma a que, quando se clica no botão “Iniciar Jogo”, o seu texto mude para “Terminar Jogo” e apresente os vários elementos com os dados do jogo, como se apresenta na figura seguinte.



Figura 6 - Jogo Iniciado

Para obter o comportamento e aspeto da figura 6, complete os seguintes passos:

- a. Implemente um *Event Listener* para o elemento **btPlay** de **forma a que quando se clica nele**:
- Se o texto for “Terminar Jogo”, deve invocar a função **stopGame** (a implementar)
  - Caso contrário deve invocar a função **startGame** (a implementar)
- b. A função **startGame** deve implementar os seguintes passos:
1. Especificar o texto “Terminar Jogo” ao botão **btPlay**
  2. O elemento de selecção do Nível (**btLevel**) deve ficar inativo
  3. Remover a class **'hide'** a todos elementos especificados com a classe **.gameStarted** (processo semelhante que fez no **reset**).
  4. Adicione a class **hide** ao elemento **message**

c. A função **stopGame** deve implementar os seguintes passos:

1. Especificar o texto “Iniciar Jogo” ao botão `btPlay`
2. O elemento de selecção do Nivel (`btLevel`) volta a ficar ativo
3. Invocar a função `Reset`.

d. Confirme, no *browser*, se o resultado final tem o comportamento pretendido. Confirme na consola que não existem erros de JavaScript.

4> Para concluir, especifique o código necessário para que, quando se clicar no panel de jogo **panelGame** não se encontre a mensagem “Clique em Iniciar Jogo!”, apresente, caso contrário, remova-a. A figura 7 apresenta o aspeto desejado. Para implementar o pretendido, apenas necessita de adicionar um *event listener* ao elemento **panelGame** e verificar se existe texto ou não.



Figura 7 – Mensagem “Clicar em Iniciar jogo”