

# *Linguagens Script*

## **<Ficha Nº2>**

---

Departamento de Engenharia Informática e de Sistemas

2021/2022

# > Jogo de Memória



# > Ficha Prática Nº2 – Dicas

---

- Declaração das variáveis globais no início do documento
- Formas de se aceder a um elemento DOM, seja para alterar o seu estado ou adicionar um *event listener*.

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

- Variáveis **que não serão alteradas**, devem ser declaradas com **const**, caso contrário, declare com **let**.
  - `const message = document.querySelector('#message');`

# > Ficha Prática Nº2 – Dicas

---

- Alteração de estilos:

- `elemento.style.display = 'none';`
- `elemento.classList.add("estilo");`

- Alteração do texto de um element:

- `elemento.textContent = 'JavaScript';`

# > Ficha Prática Nº2 – Dicas

---

## ■ *Event Listener*

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

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

# > Ficha Prática Nº2 – Dicas

---

## ■ *Event Listener*

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

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

- Existem várias formas de aplicar o método **addEventListener()**

# > Ficha Prática Nº2 – Dicas

---

- Declarando uma função externa

```
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!`);  
}
```

# > Ficha Prática Nº2 – Dicas

---

- Recorrendo, a uma função sem nome (*anonymous function*)

```
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!'));
```



# > Ficha Prática Nº2 – Dicas

---

- Recorrendo, a uma função sem nome (*anonymous function*)

```
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!')
});
```

# > Ficha Prática Nº2 – Dicas

---

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

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

- 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

*Linguagens Script*

---

# **Ficha Prática N2**

---